Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Proyecto de Fin de Master

Programa oficial de posgrado en Ingeniería Informática y de Telecomunicación

Sistema de Detección de Intrusiones Bluetooth para Terminales Móviles

Autor: Germán Retamosa de Ágreda
[german.retamosa@gmail.com]

Tutor: Jorge E. López de Vergara
[jorge.lopez_vergara@uam.es]

Madrid, Octubre de 2009

RESUMEN

La evolución de estas tecnologías está provocando un aumento significativo en el número de dispositivos por usuario y en la aparición de nuevos modelos con funcionalidades avanzadas. Estos avances están repercutiendo en la aparición de nuevos vectores de ataque que hacen vulnerables, diariamente, a un mayor número de usuarios. Recientes estudios han comprobado que un 45% de la población encuestada desconoce los riesgos de seguridad asociados a su dispositivo móvil. Además, un 50% de los encuestados aseguran haber sido víctimas de algún tipo de *malware* y, únicamente, un 20% tiene instalado un sistema de antivirus en su terminal. Estos resultados indican la precariedad de la seguridad móvil, tanto a nivel de aplicaciones comerciales como a nivel de concienciación de la sociedad. Por este motivo, la implementación de sistemas de seguridad, como el trabajo de investigación propuesto, tienen como principal objetivo concienciar a la sociedad de los riesgos de seguridad asociados a los terminales móviles, y plantear diferentes alternativas para contrarrestar estos riesgos.

El presente trabajo plantea el estudio de un sistema de detección de intrusiones Bluetooth para dispositivos móviles, sobre la plataforma de desarrollo *Windows Mobile*, que permita garantizar la confidencialidad, integridad y disponibilidad de la información personal del usuario. De esta manera, la implementación de este tipo de herramientas de seguridad proporcionará nuevos, y más seguros, niveles de confianza. Actualmente, las aplicaciones de seguridad móvil están centradas en la implementación de sistemas de antivirus. Sin embargo, estas aplicaciones dejan vulnerable al dispositivo móvil ante ataques de suplantación (*spoofing*) o denegación de servicio (*denial of service*). Por este motivo, unos de los principales objetivos del sistema propuesto consiste en la implementación de un sistema basado en firmas o basado en anomalías, mediante modelos matemáticos, que reduzcan los riesgos derivados de estos tipos de ataques.

Además, el trabajo propuesto permite la implementación de varios modos de funcionamiento, un modo estándar, un modo adaptativo y un modo basado en políticas. El modo estándar está basado en el funcionamiento de la mayoría de los sistemas de detección de intrusiones actuales, con la optimización del protocolo de comunicaciones Bluetooth. El modo adaptativo se comporta de igual manera que el modo estándar, salvo que optimiza el rendimiento en función de los perfiles detectados por el sistema periódicamente. Por último, el modo basado en políticas se comporta siguiendo un esquema en el que los eventos que se reciben en la interfaz Bluetooth son admitidos o denegados según un conjunto de políticas de negocio.

Palabras Clave: seguridad informática, comunicaciones móviles, sistema de detección de intrusiones, Bluetooth, virología móvil.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor Jorge E. López de Vergara su incondicional ayuda, y su espíritu trabajador, durante la realización del trabajo final de Master. Gracias a sus correcciones y aportaciones técnicas, sus valoraciones objetivas, y sus consejos, ha hecho posible la finalización de este trabajo de investigación.

Agradezco el esfuerzo depositado en mí por la Universidad Autónoma de Madrid, y especialmente en la beca de investigación que me ha sido concedida durante el período de realización de este trabajo, ya que sin esta ayuda no hubiera podido asistir a congresos internacionales que completaran mi formación. De igual manera, quiero agradecer la amabilidad de todos los miembros del grupo de investigación de redes y computación de alto rendimiento (HPCN) de la Universidad Autónoma de Madrid, que han hecho amenas muchas tardes de duro trabajo.

Finalmente, quiero hacer una especial mención a mi familia, a mi madre Gloria, mi padre José Luis, mi hermana Leticia, y a mi novia Susana. Todos ellos han sido mi escapatoria ante los problemas y dificultades impuestas por el fuerte trabajo llevado a cabo, y que sin ellos, todo esto hubiera sido prácticamente imposible.

Muchas gracias a todos por vuestro apoyo.

Germán Retamosa de Ágreda

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1. INTRODUCCIÓN AL PROYECTO

1.1. MOTIVACIÓN

Desde sus orígenes, los dispositivos móviles han jugado un papel esencial en las comunicaciones entre usuarios. Sin embargo, la evolución de estas tecnologías está provocando un aumento significativo en el número de dispositivos por usuario y en la aparición de nuevos modelos con funcionalidades avanzadas. Estos avances están repercutiendo en la aparición de nuevos vectores de ataque que hacen vulnerables, diariamente, a un mayor número de usuarios. Un reciente estudio llevado a cabo por la empresa de seguridad TrendMicro, [58] ha comprobado que un 45% de la población encuestada desconoce los riesgos de seguridad asociados a su dispositivo móvil. Además, un 50% de los encuestados aseguran haber sido víctimas de algún tipo de *malware* y, únicamente, un 20% tiene instalado un sistema de antivirus en su terminal. Estos resultados indican la precariedad de la seguridad móvil, tanto a nivel de aplicaciones comerciales como a nivel de concienciación de la sociedad. Por este motivo, la implementación de sistemas de seguridad, como el trabajo de investigación propuesto, tienen como principal objetivo concienciar a la sociedad de los riesgos de seguridad asociados a los terminales móviles, y plantear diferentes alternativas para contrarrestar estos riesgos.

El presente trabajo plantea el estudio de un sistema de detección de intrusiones Bluetooth para dispositivos móviles que permita garantizar la confidencialidad, integridad y disponibilidad de la información personal del usuario. De esta manera, la implementación de este tipo de herramientas de seguridad proporcionará nuevos, y más seguros, niveles de confianza. Actualmente, las aplicaciones de seguridad móvil están centradas en la implementación de sistemas de antivirus. Sin embargo, estas aplicaciones dejan vulnerable al dispositivo móvil ante ataques de suplantación (*spoofing*) o denegación de servicio (*denial of service*). Por este motivo, unos de los principales objetivos del sistema propuesto consiste en la implementación de un sistema basado en firmas o basado en anomalías, mediante modelos matemáticos, que reduzcan los riesgos derivados de estos tipos de ataques.

Un aspecto que se ha tenido tener cuenta durante la realización del proyecto, ha sido la identificación y el análisis de líneas futuras de investigación que sirvan de punto de partida para una futura tesis.

Por último, el presente capítulo está dividido, por los objetivos principales que componen el trabajo final de master, la descripción formal de las principales actividades a desarrollar y su ciclo de vida asociado. En último lugar, se encuentra definida la estructura de la memoria, junto con un breve resumen de los principales temas tratados durante el trabajo.

1.2. OBJETIVOS

Este proyecto de fin de master tiene por objeto proponer soluciones eficientes y robustas, ante las limitaciones existentes en la seguridad móvil actual. Por consiguiente, se proponen los siguientes objetivos para alcanzar dicha tarea:

- Realizar un estudio del estado del arte de los sistemas de detección de intrusiones y virología móvil, completando así los trabajos tutelados realizados durante la primera fase del master.
- Identificar los vectores de ataque existentes, que permitan evaluar las amenazas actuales en los dispositivos móviles de nueva generación.
- Realizar un estudio del estado del arte de los sistemas de comunicación Bluetooth que ayuden a mejorar el entendimiento del sistema y su funcionamiento.
- Implementar un prototipo de un sistema de detección de intrusiones móviles en tiempo real y bajo unas condiciones específicas determinadas por el dispositivo móvil y el sistema operativo utilizado.
- Analizar e identificar las diferentes líneas de investigación, mediante el estudio de nuevas problemáticas y campos de investigación, que sirvan para la realización de una futura tesis doctoral.

1.3. DESCRIPCIÓN DEL TRABAJO

En primer lugar, el siguiente apartado, mediante el estudio de los estados del arte de la virología móvil, los sistemas de detección de intrusiones y el protocolo de comunicaciones Bluetooth, ha procedido a investigar y analizar las principales necesidades y amenazas existentes en el campo de la seguridad móvil, y así, dar cierta veracidad al proceso de investigación. Posteriormente, se ha especificado el modelo de ciclo de vida utilizado para llevar a cabo el proyecto, y de acuerdo con la planificación temporal prevista en los diagramas *Gantt* incluidos en el proyecto. El modelo de ciclo de vida más adecuado para este tipo de proyecto es el modelo de desarrollo evolutivo o prototipado evolutivo. Dicha elección se ha debido a la incertidumbre generada por la implementación de un sistema dotado de gran innovación y cuyas bases no están aún asentadas de manera consistente. El trabajo se ha divido en 6 actividades detalladas a continuación:

- Actividad 1: Análisis del proyecto.
  Durante la fase de análisis se ha procedido a la especificación de los requisitos del sistema, que serán contrastados mediante la realización de diferentes revisiones al documento de análisis.
- Actividad 2: Diseño de la arquitectura.
  Una vez definida la especificación de requisitos, se ha creado el documento de diseño con los puntos necesarios para la realización de un desarrollo eficiente.
- Actividad 3: Implementación del sistema.
  La fase de implementación ha consistido en el desarrollo de los puntos tratados en el documento de diseño, mediante un lenguaje de

programación y un sistema operativo definidos en la especificación de requisitos del documento de análisis.

- Actividad 4: Fase de Validación del sistema.
  La fase de pruebas ha comprobado la funcionalidad, desde diversos escenarios y casos de uso, de los diferentes prototipos generados durante la fase de implementación, de tal manera que satisfagan la especificación de requisitos descritos en el documento de análisis.
- Actividad 5: Documentación.
  El proceso de documentación se ha realizado a lo largo del ciclo de vida del proyecto y ha incluido la generación del documento de análisis o especificación de requisitos del sistema, el documento de diseño y el plan de pruebas en la memoria final del proyecto. Opcionalmente, se ha considerado la posible creación de un manual de usuario de acuerdo con las exigencias requeridas por el proyecto.
- Actividad 6: Seguimiento y Gestión del proyecto.
  Tanto el seguimiento como la gestión de proyecto se ha realizado a lo largo del ciclo de vida del proyecto, mediante la incorporación de reuniones periódicas que evalúen su estado actual, y generen los documentos correspondientes con las planificaciones temporales ajustadas al estado real y los objetivos propuestos en cada etapa del proyecto.

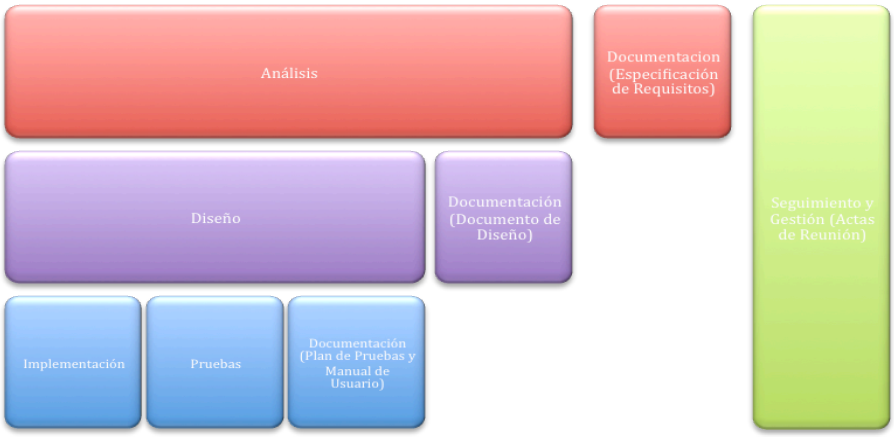

Figura 1.1: Esquema de Actividades del Proyecto

Una vez descritas las diferentes actividades que formaron el proyecto, se ha procedido a la inclusión de una planificación temporal prevista mediante los siguientes diagramas *Gantt*.

1. INTRODUCCIÓN AL PROYECTO

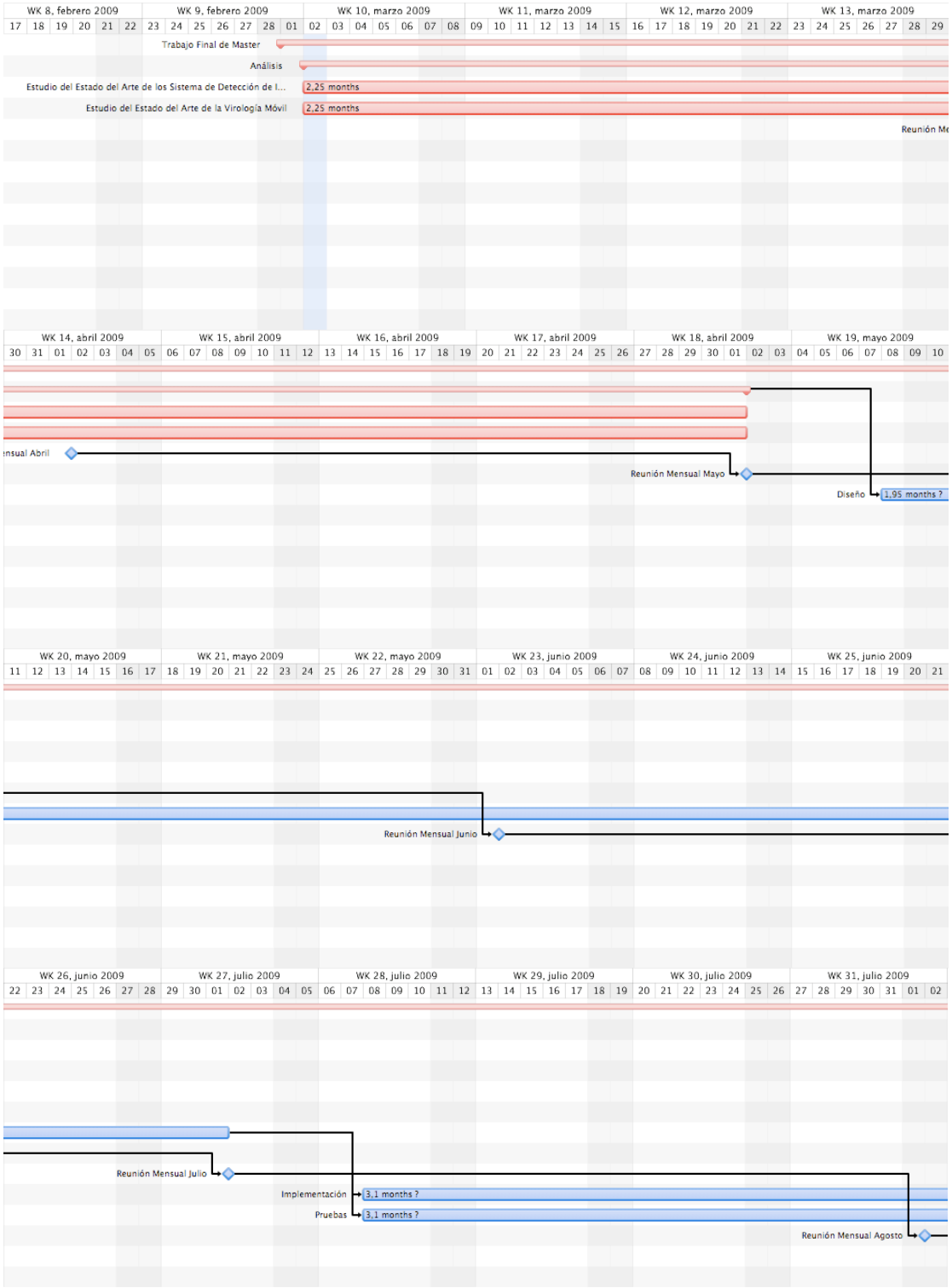

Figura 1.2: Planificación Temporal del Proyecto

1.4. ESTRUCTURA DE LA MEMORIA

En el siguiente apartado se ha comentado brevemente la estructura general y los contenidos básicos incluidos en los diferentes capítulos que componen el trabajo:

- El capítulo 1 consta de una breve introducción al proyecto, incluyendo la motivación necesaria para su realización, los objetivos propuestos y una descripción general que determine el ciclo de vida empleado. Además, el presente capítulo ha proporcionado una estimación temporal del tiempo necesario para cada actividad.
- El capítulo 2 presenta una introducción a los sistemas de detección de intrusiones mediante el estudio de su estado del arte. En este capítulo se ha analizado, con especial interés, los diferentes tipos y modelos de sistemas de detección de intrusiones existentes en la actualidad.
- El capítulo 3 presenta una introducción a la virología móvil mediante el estudio de su estado del arte. En el siguiente capítulo, se ha estudiado los vectores de ataque más conocidos con independencia del sistema operativo utilizado. Sin embargo, una vez descritos los conceptos generales se ha analizado un único sistema operativo que permita focalizar los estudios a un modelo de seguridad concreto.
- El capítulo 4 presenta una introducción a los sistemas de comunicación Bluetooth mediante el estudio de su estado del arte. En el siguiente capítulo se ha estudiado la arquitectura Bluetooth, compuesta por dos

bloques principales, los componentes centrales y los servicios o perfiles de comunicación.
- El capítulo 5 detalla las cuestiones relacionadas con la fase de análisis del proyecto, partiendo de los estudios previos de los sistemas de detección de intrusiones, la virología móvil y el protocolo de comunicaciones Bluetooth.
- El capítulo 6 detalla las cuestiones relacionadas con la fase de diseño del proyecto, especificando la arquitectura general del sistema y los métodos utilizados para su realización.
- El capítulo 7 ha atendido las diferentes cuestiones relacionadas con la fase de validación y los entornos utilizados para su realización.
- El capítulo 8 consta de las conclusiones obtenidas a partir del trabajo realizado, incluyendo las diferentes líneas de investigación y trabajos futuros a realizar.
- El capítulo 9 consta de la referencia bibliográfica utilizada.

1.5. CONCLUSIONES

Los puntos tratados durante el presente capítulo tienen por objeto servir de guía de referencia ante los futuros capítulos que han sido explicados durante el trabajo de investigación. La motivación del trabajo permite justificar la necesidad de investigación del tema propuesto. Además, la descripción del trabajo permite realizar un seguimiento de las tareas llevadas a cabo durante el proyecto mediante los diagramas *Gantt* incluidos.
El siguiente capítulo introduce el estudio de los estados del arte, sobre los sistemas de detección de intrusiones en primer lugar, que sirvan de preparaciónn ante el posterior diseño de la plataforma propuesta.

2. SISTEMAS DE DETECCIÓN DE INTRUSIONES

La década de los años ochenta fue el inicio de las primeras investigaciones en la detección de intrusiones. Estas investigaciones eran concebidas como la fusión entre la auditoria del procesamiento de los datos electrónicos y la auditoria de seguridad mediante procesos de reconocimiento de patrones y procesos estadísticos optimizados. A partir de los siguientes estudios, se proporcionó una definición formal del concepto de la detección de intrusiones o IDS (*Intrusion Detection System*), como el proceso reactivo de monitorización de eventos que ocurren en un sistema o red, mediante el análisis de las señales relacionadas con sus correspondientes problemas de seguridad. En primer lugar, la auditoria de seguridad define el proceso de generación, registro y revisión de un conjunto de eventos ocurridos en un sistema ordenado cronológicamente. Este proceso está basado en un conjunto en un conjunto de reglas o políticas que toman decisiones sobre el sistema y se fundamenta en los siguientes objetivos [25]:

- Reconstrucción de eventos.
- Evaluación de riesgos.
- Asignar y mantener una responsabilidad personal de actividades en el sistema.
- Monitorizar áreas problemáticas.
- Recuperación eficiente ante fallos.
- Detectar uso inapropiado del sistema.

En segundo lugar, la auditoria del procesamiento de los datos electrónicos permite una monitorización interna del sistema, respecto a la auditoria perimetral proporcionada por el caso anterior. Sin embargo, este caso requiere su conocimiento interno.

La Figura 2.1 muestra la arquitectura genérica seguida por la mayoría de los sistemas de detección de intrusiones. Esta arquitectura, definida por los grupos de trabajo de referencia CIDF (*Common Intrusion Detection Framework*) [27] e IDWG (*Intrusion Detection Working Group*), se compone de cuatro bloques principalmente: bloques E o eventos, bloques D o bases de datos, bloques A o análisis, y bloques R o respuesta. Sin embargo, algunos esquemas de representación incluyen, en algunas ocasiones, un quinto elemento llamado SSO (*Site Security Officer*), encargado de controlar y administrar los recursos del sistema. Los bloques E son los encargados de obtener toda la información disponible del sistema, para posteriormente proporcionar toda esta información a los bloques A y almacenarla en los bloques D. Los bloques D son los encargados de almacenar la información del sistema y proporcionársela a los bloques A cuando sea necesario. Los bloques A son los encargados de analizar la información, determinar si la información proporcionada se considera una posible amenaza y proporcionar una

respuesta a los bloques R. Los bloques R son los encargados de realizar las acciones solicitadas por los bloques A. El intercambio de información entre los diferentes bloques que constituyen la arquitectura sigue un protocolo de referencia, IDXP (*Intrusion Detection Exchange Protocol*, RFC 4767), y una estructura de datos bien definida, IDMEF (*Intrusion Detection Message Format*, RFC 4768). Finalmente, la arquitectura descrita posee una serie de principios y responsabilidades descritos a continuación:

- Asignación de responsabilidades.
- Evaluación del riesgo o daño.
- Recuperación del riesgo o daño.

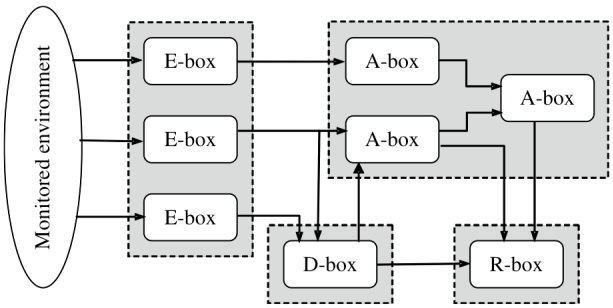

Figura 2.1: Arquitectura CIDF [26]

Por último, el presente capítulo está compuesto por la definición de un esquema de categorización y clasificación de los principales sistemas de detección de intrusiones. Posteriormente, se han escogido algunos de los ejemplos más representativos, tales como *IDES*, *Snort*, *Tripwire* o *Prelude*, correspondientes a cada uno de los diferentes tipos de sistemas, y asi, tener una visión global del concepto de detección de intrusiones.

2.1. CATEGORIZACIÓN

La categorización de los sistemas de detección de intrusiones va a consistir en el estudio de cuatro factores principales que constituyen estos sistemas: las estrategias de monitorización, los tipos de análisis, los tipos de temporización y los esquemas de control.

Los **esquemas de control** muestran la arquitectura seleccionada para la gestión y el mantenimiento del sistema de detección. Principalmente, estos esquemas son de carácter distribuido o centralizado. En primer lugar, los sistemas distribuidos constan de varios puntos de gestión y una alta tolerancia a fallos. En segundo lugar, los sistemas centralizados constan de un único punto de control y una mayor simplicidad de implementación y mantenimiento. La elección de este tipo de esquemas está basada en la enumeración de una serie de factores detallados a continuación:

- Sensibilidad del sistema.
- Naturaleza del sistema.
- Naturaleza de las políticas de seguridad.
- Nivel de amenaza.

Los **tipos de temporización** indican la fase temporal donde se realiza el proceso de análisis de la información por los bloques A. Los tipos de temporización existentes son en tiempo real o mediante intervalos (*batch*). En primer lugar, los sistemas en tiempo real requieren un alto coste computacional y poseen una mayor probabilidad de generar falsas aceptaciones, pero devuelven una respuesta inmediata al usuario y cubren un mayor número de intrusiones. En segundo lugar, los sistemas por intervalos requieren una menor carga computacional del sistema, pero el retardo generado durante su análisis provoca la reducción del rango de intrusiones detectadas.
Los **tipos de análisis** determinan las técnicas utilizadas para procesar la información obtenida por los bloques E y generar una respuesta concreta para realizar la contramedida correspondiente. Los principales tipos de análisis son los basados en firma o en anomalías.
En primer lugar, las técnicas basadas en firma o SIDS (*Signature-based IDS*) utilizan métodos de reconocimiento de patrones (*pattern-matching*) para identificar los posibles ataques en el sistema. Estas técnicas almacenan un número finito de firmas, cada una representando un tipo de ataque, en los bloques D. Por consiguiente, el desarrollo de estas firmas es un proceso fundamental para obtener un buen rendimiento del sistema. A continuación, el siguiente ejemplo describe el proceso seguido durante la creación de firmas [28]:

- Trin00 (http://www.snort.org/snort-db/sid.html?sid=223)
  - Creación de un identificador único
    GEN:SID 1:223
  - Creación de un mensaje identificativo
    DDOS Trin00 Daemon to Master PONG message detection.
  - Creación de la regla que identifica el ataque
    alert udp $EXTERNAL_NET any -> $HOME_NET 31335
    (msg:"DDOS Trin00 Daemon to Master PONG message
    detected"; content:"PONG"; reference:arachnids,187;
    classtype:attempted-recon; sid:223; rev:3

Por último, la madurez de estas técnicas se ve representada en la mayor utilización respecto a los modelos basados en anomalías, pero requiere conocer a priori los posibles ataques.
En segundo lugar, las técnicas basadas en anomalías o AIDS (*Anomaly-based IDS*) utilizan una amplia variedad de modelos matemáticos de clasificación para reconocer los posibles ataques al sistema. Los modelos matemáticos de clasificación se engloban en los siguientes grupos [26]:

- Métodos estadísticos.
  - Monovariable
    Modelización de variables aleatorias *gaussianas*.
  - Multivariable
    Modelización mediante la correlación entre dos o más métricas.

- Series Temporales
  Modelización mediante el cálculo probabilístico de ocurrencia entre eventos.

- Sistemas Expertos.
- Aprendizaje Automático.

Los métodos estadísticos están basados en la estimación de similitud entre una medida de referencia, previamente entrenada, y otra medida tomada en el momento de la comprobación. Si el factor de similitud es menor que un determinado umbral previamente establecido, la muestra se considerada como un posible ataque y se realiza la correspondiente contramedida, en caso contrario, no se realiza ninguna acción sobre el sistema. Los sistemas expertos están basados en la extracción de características entre una medida de referencia y una medida en el momento de la comprobación, para la posterior toma de decisiones a partir de inferencia lógica. Estos sistemas requieren un proceso de construcción manual y reducen el número de falsos positivos respecto al modelo anterior.

A continuación, la Figura 2.2 muestra un esquema de los diferentes sistemas de aprendizaje automático, junto con sus principales ventajas e inconvenientes. Además, la Figura 2.3 muestra un resumen de los principales ejemplos de sistemas basados en firma y basados en anomalías.

Redes Bayesianas	• Interdependencia variables • Predicción de eventos	• Alto coste computacional • Dependencia del sistema
Modelos de Markov	• Independencia del sistema	
Redes Neuronales	• Flexibilidad y Adaptabilidad	• Ausencia de modelo descriptivo.
Lógica Difusa	• Efectivo frente *port scans*	• Alto consumo recursos • Poco aceptado
Algoritmos Genéticos	• Robustez y flexibilidad	• Alto consumo recursos
Clustering	• Coste computacional bajo	

Figura 2.2: Esquema de técnicas de aprendizaje automático [26]

Sistemas basados en detección de Anomalías	Autoaprendizaje	Series no temporales	Modelado de reglas	W&S
			Estadística Descriptiva	IDES, NIDES, EMERALD
		Series Temporales	ANN	Hyperview
	Programado	Estado descriptivo	Estado simple	MIDAS, NADIR
			Reglas simples	NSM
			Umbral	ComputerWatch
		Denegación por defecto	Modelado de series de estado	DPEM, Janus, Bro
Sistemas basados en detección por Firma	Autoaprendizaje	Selección autómatica	Ripper	
	Programado	Modelado por Estado	Transición de Estados	USTAT
			Redes de Petri	IDIOT
		Sistema Experto	NIDES, EMERALD, DIDS	
		Coincidencia Textuales	NSM	
		Reglas Simples	NADIR, ASAX, Bro	

Figura 2.3: Resumen de los principales SIDS y AIDS

Finalmente, el último de los factores que se van a estudiar en el siguiente apartado son las **estrategias de monitorización**. Estas estrategias definen los principales tipos de sistemas de detección de intrusiones, que se detallan a continuación:

- IDS basado en el *host* (*HIDS, Host Intrusion Detection System*).
- IDS basado en el segmento red (*NIDS, Network Intrusion Detection System*).
- IDS basado en una aplicación (*APIDS, Application Intrusion Detection System*).
- IDS híbridos.

Los sistemas HIDS monitorizan el comportamiento interno del sistema, mediante el uso de parámetros como los recursos del sistema. Sin embargo, la realización de estas técnicas requiere un conocimiento interno del sistema. Ejemplos de este tipo de sistema son *Tripwire*[31] y *OSSEC* [57].

Los sistemas NIDS monitorizan un segmento de red, mediante la utilización de parámetros como el protocolo de comunicación utilizado o las direcciones de origen y destino de los paquetes. Ejemplos de este tipo de sistema son *Bro* [56] y *Snort* [29].

Los sistemas APIDS monitorizan una aplicación concreta mediante el análisis de sus entradas y salidas. Una situación de ejemplo de este tipo de sistemas consistiría en el análisis por parte de un servidor web y un gestor de bases de datos sobre el protocolo SQL (*Structured Query Language*).

Por último, los sistemas híbridos consisten en las fusión de los sistemas HIDS y NIDS mediante el análisis de los factores más significativos de ambos, y proporcionando un contexto de protección más amplio y un nivel de seguridad más elevado. Un ejemplo de estos sistemas híbridos es DIDS mediante la integración de los servicios basados en *host* por *Haystack*, y los servicios basados en el segmento de red por *NSM*.

2.2. SISTEMAS DE DETECCIÓN DE INTRUSIONES

La madurez de los sistemas de detección de intrusiones ha generado un amplio rango de tipos de sistemas de detección en función del proceso de análisis sobre la información recopilada del sistema. A continuación, se describen algunos ejemplos de sistemas de detección de intrusiones con sus respectivas técnicas de análisis. En primer lugar, el criterio de selección seguido se ha basado en un orden cronológico con la incorporación del precursor de los sistemas de detección de intrusiones, *IDES* [30]. Posteriormente, se han incluido aquellos ejemplos más representativos, acorde con los diferentes modelos de sistemas de detección de intrusiones más significativos en la actualidad, NIDS (*Snort* [29], HIDS (*Tripwire*[31]) e IDS híbridos (*Prelude*[33]).

2.2.1. IDES

IDES (*Intrusion Detection Expert System*)[30] fue el primer sistema de detección de intrusiones en tiempo real basado en anomalías, y el precursor de los sistemas expertos. Este sistema de detección de intrusiones de red, creado por Dorothy E. Denning entre 1984 y 1986, constituyó la base de los sistemas de detección actuales y sirvió de punto de partida para la creación de otros tipos de sistemas de detección como *NSM*, *NADIR* o *Haystack*.
La Figura 2.4 muestra la arquitectura híbrida de *IDES* formada por un detector de anomalías y un sistema experto. En primer lugar, el detector de anomalías está basado en estructuras de datos denominadas perfiles (*profiles*), que usan métricas y modelos estadísticos para caracterizar el uso anormal del sistema. En segundo lugar, el sistema experto utiliza un conjunto de reglas de actividad para detectar los ataques y reducir los riesgos producidos de ataques a lo largo del tiempo. Finalmente, el sistema de detección de intrusiones *IDES*[30] implantó las bases de los sistemas actuales y se mantiene hoy en día en constante evolución mediante su evolución de nueva generación NIDES (*Next-Generation IDES*).

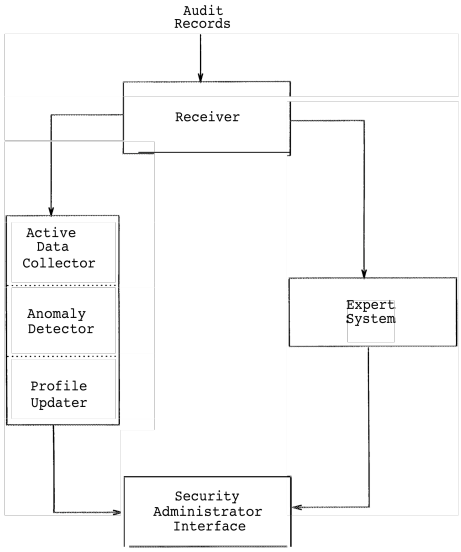

Figura 2.4: Arquitectura IDES [30]

2.2.2. SNORT

Snort[29] consiste en un sistema de detección de intrusiones de red ligero y basado en tiempo real. Sin embargo, este sistema posee cierta madurez en el panorama de la seguridad actual, ya que su historia se inicia en noviembre de 1998, cuando Marty Roesch creó su inmediato antecesor, APE. La principal técnica de análisis utilizada consiste en el reconocimiento de firmas de ataques, aunque utiliza otras técnicas como la inferencia de reglas mediante el uso de sistemas expertos. Además, este sistema aporta funcionalidades adicionales como la captura (*sniffing*) y el registro (*logging*) de paquetes.
La Figura 2.5 muestra la arquitectura seguida por *Snort*.

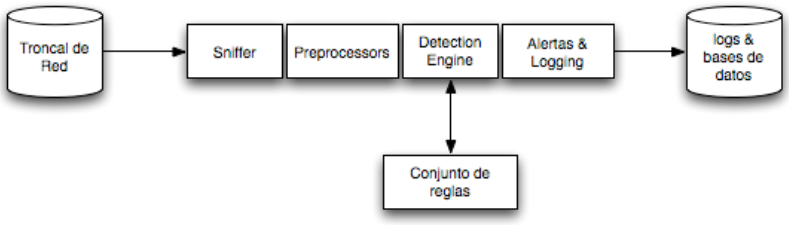

Figura 2.5: Arquitectura *Snort* [29]

Esta arquitectura está compuesta por un conjunto de elementos claramente diferenciados. En primer lugar, el *sniffer* es el componente encargado de capturar toda la información de red posible, para procesarla posteriormente mediante los diferentes clasificadores (*preprocessors*) existentes en el sistema. Además, este sistema posee tantos clasificadores como atributos de red se deseen analizar, como por ejemplo los puertos de origen y destino o las llamadas a procedimientos remotos. Tras la clasificación de la información, el sistema comienza el proceso de análisis (*detection engine*) dirigido por un sistema experto formado por reglas. Cada una de estas reglas se encuentra

agrupada por categorías y tipos, y están compuestas por los siguientes elementos:

- Cabecera (*Header*)
  - Acción a realizar.
  - Protocolo de comunicación.
  - IP origen.
  - IP destino.
  - Puerto origen.
  - Puerto destino.
- Información a analizar (*Body*).

Por último, toda esta información se almacena en las correspondientes bases de datos para su posterior utilización. Finalmente, este sistema se caracteriza por la simplicidad de implementación y la rapidez de procesamiento de la información de red. Además, las técnicas de análisis utilizadas reducen el número de falsos positivos y negativos.

2.2.3. TRIPWIRE

Tripwire[31] es actualmente considerado, junto con *OSSEC*, uno de los principales sistemas de detección de intrusiones basados en *host*. Este sistema de detección en tiempo real fue diseñado en 1992, y permite monitorizar cambios a nivel del sistema de archivos. Además, este proceso de detección de cambios está basado en la comparación por *hashes* de objetos del sistema de archivos, que reduce la problemática del almacenamiento.

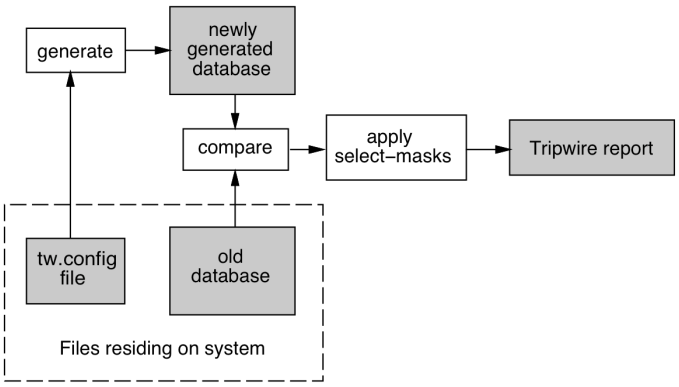

Figura 2.6: Esquema operativo *Tripwire* [31]

Como se puede observar en la Figura 2.6, el flujo de operaciones seguido por este sistema consta de un conjunto de entradas a monitorizar como el fichero de configuración (*tw.config*), que determina los objetos del sistema de ficheros a monitorizar, y la base de datos con los *hashes* correspondientes. Además, una de las principales características de este sistema consiste en la reutilización de componentes, por la que el fichero de configuración puede estar instalado en diferentes máquinas, mientras que la base es datos es generada internamente. El modelo de creación de *hashes* está dirigido por políticas de seguridad locales, permitiendo la asignación de múltiples tipos de *hashes* por objeto del

sistema de ficheros, y proporcionando un factor de redundancia en la integridad de la información. Algunos ejemplos de estas implementaciones se detallan a continuación:

- MD5, MD4, MD2, SHA.
- CRC-32 (compatibilidad POSIX 1003.2).
- CRC-16 (compatibilidad CCITT).

Además, el proceso de monitorización consta de un conjunto de parámetros característicos como el número de inodo, el número de enlaces, el UID, el GID, el tamaño del fichero, o la marca de tiempo (*timestamp*). Finalmente, este sistema aporta factores como la portabilidad, la escalabilidad y la facilidad de configuración, junto con un conjunto de servicios adicionales como el aseguramiento de la integridad de la información y una gestión de cambio.

2.2.4. PRELUDE

Prelude[33] es un sistema de detección de intrusiones híbrido bajo una licencia GPL (*General Public License*). Este sistema, también llamado USIM (*Universal System Information Management*), comparte las principales características de los sistemas de detección basados en *host* y basados en red. La normalización por el estándar de facto IDMEF aporta una interoperabilidad respecto a otros sistemas. No obstante, este sistema aporta otras muchas características como por ejemplo, modularidad, extensibilidad y una alta disponibilidad de los servicios existentes. La Figura 2.7 muestra el flujo de trabajo seguido por *Prelude* donde la recolección de información es realizada por los múltiples sensores heterogéneos que conforman la arquitectura.

Figura 2.7: Flujo de Trabajo *Prelude* [33]

El proceso de normalización sigue las reglas establecidas por el estándar IDMEF y las fases de ordenación y agregación forman parte del procesamiento de datos llevado a cabo por los *managers* de la arquitectura. El proceso de correlación es altamente modificable y está divido en un conjunto de procesos de correlación en tiempo real basados en reglas y basados en técnicas estadísticas. Por último, las acciones desencadenantes del proceso de correlación generan un conjunto de alertas que generan informes altamente modificables y en diversos formatos.

La Figura 2.8 muestra un diagrama con la arquitectura genérica seguida por *Prelude*.

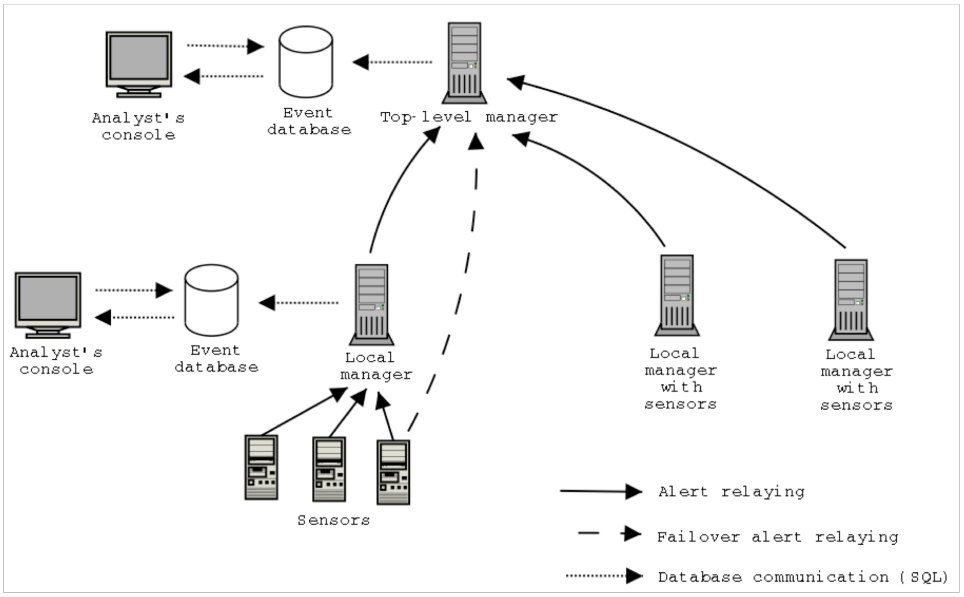

Figura 2.8: Arquitectura genérica Prelude [32]

El carácter distribuido de su arquitectura permite establecer un alcance global. En una primera aproximación, este sistema consta de dos bloques principales o *daemons*, detección y generación de informes. Sin embargo, una visión más detallada de la arquitectura permite diferenciar entre tres tipos de componentes, los sensores, los *managers* y el *frontend*. Los sensores son los encargados de recolectar la información del sistema mediante proceso *agentless* o independiente de los diferentes fabricantes y formatos de información. Los *managers* son los encargados del procesamiento de la información recolectada por los sensores. Este proceso está dividido en la clasificación y el filtrado de los datos con sus correspondientes niveles (recolección, procesamiento y visualización), el transporte seguro entre los diferentes módulos que componen el sistema y el almacenamiento centralizado de la información mediante sus correspondientes políticas de retención de datos. Por último, el *frontend* es el encargado de la visualización en tiempo real mediante su correspondiente interfaz de usuario. Además, este esquema de visualización añade unos módulos de administración y autenticación que garanticen unos niveles de servicio adecuados. Finalmente, los sistemas de detección híbridos, como *Prelude*, aportan técnicas más completas, eficientes y adaptadas a los nuevos retos que plantean las tecnologías de la información.

2.3. CONCLUSIONES

Los puntos tratados durante el presente capítulo tienen por objeto introducir el concepto de la detección de intrusiones. Esta introducción consta de una parte teórica, mediante el estudio de los elementos que permiten categorizar y clasificar cada sistema, y una parte práctica, mediante el estudio del

funcionamiento de los ejemplos más representativos. Por este motivo, gracias al estudio realizado durante el presente capítulo, el diseño del sistema propuesto está fundamentado en unas bases sólidas procedentes de sistemas comerciales con cierta madurez.
El siguiente capítulo sigue con la línea introductoria comenzada por el presente capítulo, mediante el estudio del estado del arte de la virología móvil, que permita reconocer las principales amenazas en los terminales móviles.

3. VIROLOGÍA MÓVIL

La actual evolución de los dispositivos móviles ha venido determinada por el creciente número de funcionalidades requeridas por los usuarios. Por este motivo, este notable aumento de los teléfonos móviles en el mercado de las telecomunicaciones, han supuesto la aparición de una nueva era tecnológica. La creación de estándares de comunicación inalámbricos como el IEEE 802.11 (*Wi-Fi* [1]) o el IEEE 802.15.1 (*Bluetooth* [35]) entre otros muchos, presentan la necesidad de una alta disponibilidad y calidad en los servicios existentes. Sin embargo, esta revolución no sólo aporta mejoras y ventajas para el usuario, sino que también lleva consigo una serie de debilidades y problemáticas que deben ser tenidas en cuenta para garantizar unos niveles de seguridad mínimos. La aparición de la virología en el contexto de las comunicaciones móviles ha sido una de las problemáticas desencadenadas por esta revolución y que ha sido debidamente analizada mediante el estudio de su estado del arte.

Por último, el presente capítulo está compuesto por la descripción de los principales sistemas operativos móviles y sus correspondientes plataformas de seguridad [55]. Posteriormente, se ha procedido a una definición teórica de la virología móvil, mediante la descripción de su morfología y taxonomía, ambos utilizados frecuentemente por las empresas de antivirus.

3.1. SISTEMAS OPERATIVOS MÓVILES

Una de los principales problemas existentes en las comunicaciones móviles es la fragmentación móvil. Debido a dicha fragmentación, no existe un estándar de facto maduro y robusto, que regule aspectos como la diversidad de sistemas operativos móviles y sus correspondientes implementaciones. Sin embargo, organizaciones sin ánimo de lucro, como la OMTP (*Open Mobile Terminal Platform* [3]) o la OMA (*Open Mobile Alliance* [54]), empiezan a trabajar en el establecimiento de estándares robustos y maduros que minimicen todos los posibles problemas derivados de la evolución de las comunicaciones móviles. Además, la problemática generada por la fragmentación móvil también afecta de manera directa a las fases de diseño e implementación de la virología móvil, ya que permiten la diversificación de los desarrollos correspondientes a los diferentes sistemas operativos. Esta característica posee tanto aspectos positivos como negativos, ya que las plataformas más vulnerables han sido las más afectadas por los ataques, aunque no han sido propagadas a otras plataformas debido al aislamiento existente entre los sistemas operativos.

El siguiente estudio analizará los sistemas operativos móviles más relevantes en la actualidad, *Symbian OS* [4], *Windows Mobile* [5], *iPhone OS* [6], *Blackberry OS* [7], *Palm OS* [8] y *Android* [9], prestando especial interés a sus plataformas de seguridad, y permitiendo así orientar el estado del arte a los diferentes casos de virología existentes. Cada uno de los sistemas operativos citados se

encuentra englobado en una categoría concreta con sus características correspondientes. El esquema de clasificación utilizado consta de las siguientes categorías [10] [55]:

- Plataformas Abiertas:
  Este tipo de sistema operativo móvil es el caso más común y expandido del mercado de las telecomunicaciones. Permite el acceso a las diferentes interfaces nativas del sistema mediante su correspondiente SDK (*Software Development Kit*) de dominio público. Sin embargo, a pesar de todas las ventajas, este tipo de sistemas operativos son los más atacados, debido al número de interfaces que el usuario puede utilizar de manera ilícita. Los ejemplos más significativos de este tipo de sistemas son *Symbian OS*, *Windows Mobile* y *Palm OS*.
- Entornos de Ejecución basados en Capas:
  Este tipo de sistema operativo es considerado un entorno cerrado, denominado *sandboxed environment*, que únicamente permite el acceso a las interfaces nativas en situaciones muy controladas, y rechazando el acceso a dichas interfaces para el resto de los casos. Además, este tipo de sistema proporciona su correspondiente SDK de dominio público para el desarrollo de aplicaciones. Los ejemplos más relevantes de este tipo de sistemas son *iPhone OS*, *Android* y *Blackberry OS*.
- Plataformas Cerradas:
  Este tipo de sistema operativo rechaza cualquier intento de acceso a las interfaces nativas del sistema, y nunca proporciona un SDK para el desarrollo de aplicaciones, ya que éstas son diseñadas e implementadas por el propio fabricante y sus correspondientes proveedores asociados. Un ejemplo de este tipo de sistemas es el sistema operativo propietario de LG [11].

Tras la clasificación de los diferentes sistemas operativos móviles, a continuación se analizará con especial interés, y de manera individualizada, las plataformas abiertas y los entornos de ejecución basados en capas debido a su relevancia y destacable presencia en el actual mercado de las telecomunicaciones.

3.1.1. SYMBIAN OS

Symbian OS surge de la alianza de varias multinacionales de telefonía móvil, actualmente denominada *Symbian Foundation*. Sin embargo, sus orígenes provienen de una pequeña empresa de telefonía móvil llamada *Psion*, y su principal sistema operativo EPOC32. Este sistema monousuario, optimizado para procesadores ARM (*Advanced RISC Machines*), se basaba en la multitarea por reemplazo y el acceso protegido a memoria, con el objetivo de separar la información del usuario de los datos del sistema operativo. Con el paso del tiempo, dicho sistema fue evolucionando y madurando de manera significativa hasta alcanzar a ser el sistema operativo móvil más utilizado y conocido. Actualmente, *Symbian OS* es un sistema operativo monousuario y multitarea basado en el microkernel de tiempo real EKA2 (*EPOC Kernel Architecture*

32bits). Además, este sistema se encuentra optimizado para un bajo consumo de energía y un alto rendimiento en el acceso a memorias ROM (*Read-only Memory*) [12].

Tras una breve introducción del sistema operativo *Symbian OS*, se va a proceder al análisis de la plataforma de seguridad que implementa. La Figura 3.1 muestra un diagrama con los principales componentes de dicha arquitectura, los cuales se detallan a continuación [10] [55]:

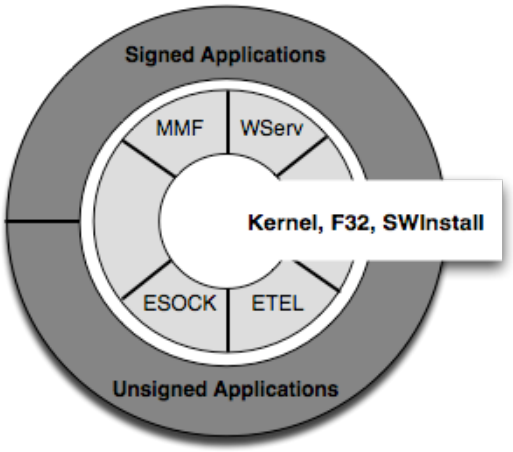

Figura 3.1: Arquitectura de Seguridad de Symbian OS [10]

- TCB (*Trusted Computing Base*).
- TCE (*Trusted Computing Environment*).
- Software firmado.
- Software no firmado.

TCB es la pieza central de la plataforma de seguridad de *Symbian OS* y, por lo tanto, el componente más seguro encargado de controlar los mecanismos de seguridad más internos del sistema. Además, este componente se encuentra formado por un conjunto de elementos como el *kernel*, el *file server* (*f32*) y el *software installer* (*SWInstall*). El *kernel* es el encargado de asignar los permisos y privilegios correspondientes durante la creación del proceso. El *file server* (*f32*) es el encargado de cargar el código del programa durante la creación del proceso. Finalmente, el *software installer* (*SWInstall*) es el encargado de instalar y validar los programas de instalación SIS (*Symbian OS Install Script*). **TCE** consiste en un entorno de ejecución formado por un conjunto de servidores con funcionalidades concretas. Cada uno de estos servidores posee funcionalidades independientes al resto y un conjunto de permisos necesarios para acceder a sus correspondientes interfaces. Por último, el **software firmado** es un conjunto de aplicaciones con derechos de acceso a las diferentes interfaces del sistema, y el **software no firmado** son aquellas aplicaciones sin derechos de acceso. El flujo de trabajo seguido durante el proceso de instalación verifica en primer lugar si la aplicación se encuentra firmada. En caso afirmativo, los componentes TCB y TCE clasificarán la firma o firmas que posean la aplicación, asignando sus valores internos UID (*User ID*), VID (*Vendor ID*) y SID (*Signature ID*). En caso contrario, no se asignarán permisos.

Finalmente, la arquitectura de seguridad de *Symbian OS* es una de las más robustas y maduras analizadas en este estudio. Además, esta robustez se traduce en una considerable reducción del número de vulnerabilidades y ataques sufridos al sistema.

3.1.2. WINDOWS MOBILE

Windows Mobile es, técnicamente, una versión del sistema operativo *Windows Compact Edition CE*. Este sistema, optimizado para procesadores ARM, está basado en la API (*Application Program Interface*) de *Win32* y el *.NET Compact Framework*. Sin embargo, la plataforma proporcionada por *Windows Mobile* no es homogénea, ya que ésta se encuentra compuesta por tres tipos de sistemas diferentes en función del tipo de dispositivo al cual van dirigidos. La versión *Standard SDK* va dirigida a los *smartphones*, la versión *Professional SDK* va dirigida a las *PDAs* con teléfono y, por último, la versión *Classic SDK* va dirigida a las *PDAs* sin teléfono. Cada uno de estos modelos posee un nivel de seguridad diferente, que define el flujo de trabajo seguido durante la instalación de aplicaciones en el teléfono móvil. De esta manera, la plataforma de seguridad está formada por la suma del conjunto de políticas de seguridad, el conjunto de roles y certificados del sistema. Sin embargo, antes de describir los niveles de seguridad existentes en la plataforma de seguridad, es necesario

enumerar y describir los principales permisos o privilegios existentes [13] [14] [55]:

- Modo *kernel* o privilegiado:
  Este tipo de permisos permite el acceso de lectura y escritura al sistema de ficheros.
- Modo usuario o normal:
  Este tipo de permisos permite el acceso de lectura y escritura a las zonas habilitadas para el usuario y rechaza el acceso al resto de zonas.
- Bloqueado:
  Este tipo de permisos rechaza cualquier tipo de acceso al sistema de ficheros.

Los niveles de seguridad existentes en *Windows Mobile* se detallan a continuación [13] [14]:

- Sin Seguridad
  Este nivel de seguridad es utilizado por todas las plataformas de desarrollo, y es alcanzado cuando todas las medidas de seguridad que posee el dispositivo, son desactivadas. En concreto, este nivel de seguridad se utiliza principalmente en terminales de homologación.
- Modelo monocapa:
  Este nivel de seguridad es utilizado por las plataformas *Professional SDK* y *Classic SDK*. Además, el flujo de trabajo seguido durante la instalación de aplicaciones verifica la existencia de certificados en la aplicación. En caso afirmativo, el sistema asigna acceso completo al sistema si el certificado es conocido en sus almacenes de certificados, o aplica las políticas de seguridad existentes para certificados desconocidos. En caso contrario, se aplica las políticas de seguridad existentes para aplicaciones sin certificados.
- Modelo bicapa:
  Este nivel de seguridad es utilizado por la plataforma *Standard SDK*. Además, el flujo de trabajo seguido durante el proceso de instalación es análogo al modelo monocapa, salvo que la asignación de permisos cuando la aplicación contiene un certificado conocido por el sistema, se realiza en función del almacén de certificados correspondiente. Por este motivo, la principal diferencia entre el modelo monocapa y el bicapa se encuentra los niveles de validación. El primer modelo posee un único nivel, mientras que el segundo modelo posee dos niveles con estructura más compleja.
- Restricción por *Mobile2Market*:
  Mobile2Market es el programa de certificación de *Microsoft* para aplicaciones de movilidad. Por consiguiente, este nivel de seguridad comprueba, a nivel de aplicación, la existencia del programa correspondiente en el entorno *Mobile2Market* con el objetivo de aplicar las políticas correspondientes para esta restricción.
- Restricción total
  Este nivel de seguridad es utilizado por todas las plataformas de desarrollo, y es alcanzado cuando todas las medidas de seguridad del

dispositivo se encuentran activadas y en su nivel más estricto. Por este motivo, este nivel de seguridad es raramente alcanzado, salvo en terminales bloqueados, ya que reduce en gran medida la experiencia de usuario.

Finalmente, la arquitectura de seguridad de *Windows Mobile* posee el sistema de políticas de seguridad, roles y certificados más robusto de las plataformas analizadas. Sin embargo, la administración de estos componentes es bastante vulnerable ante una gran variedad de ataques.

3.1.3. PALM OS

Palm OS es un sistema operativo monousuario y multitarea diseñado por *Palm Inc.* para *PDAs* y *smartphones* con procesadores ARM. Además, pese a la protección de memoria proporcionada por este sistema operativo, que separa la información personal del usuario de los datos del sistema, contempla una serie de debilidades [19]. El acceso a todas las interfaces del sistema, y captura de cualquier llamada al sistema operativo por parte de aplicaciones instaladas sin privilegios, son claros ejemplos de este tipo de vulnerabilidades. Sin embargo, las nuevas y futuras tendencias de *Palm Inc.* sobre sus futuros sistemas operativos, como el reciente Palm WebOS [8], modifican la progresión seguida durante estos últimos años, y afectan directamente sobre la arquitectura de seguridad existente. Este nuevo sistema operativo monousuario y multitarea basado en el *kernel* de Linux 2.6, se basa en un conjunto de tecnologías como XHTML (*eXtensible Hypertext Markup Language*), JavaScript y CSS (*Cascading Style Sheets*). Por consiguiente, el uso de este nuevo tipo de tecnologías, y su correspondiente entorno de desarrollo (*Mojo*), motivan la aparición de nuevos vectores de ataque orientados a la tecnología utilizada.
La Figura 3.2 muestra la arquitectura genérica seguida por *Palm WebOS*.

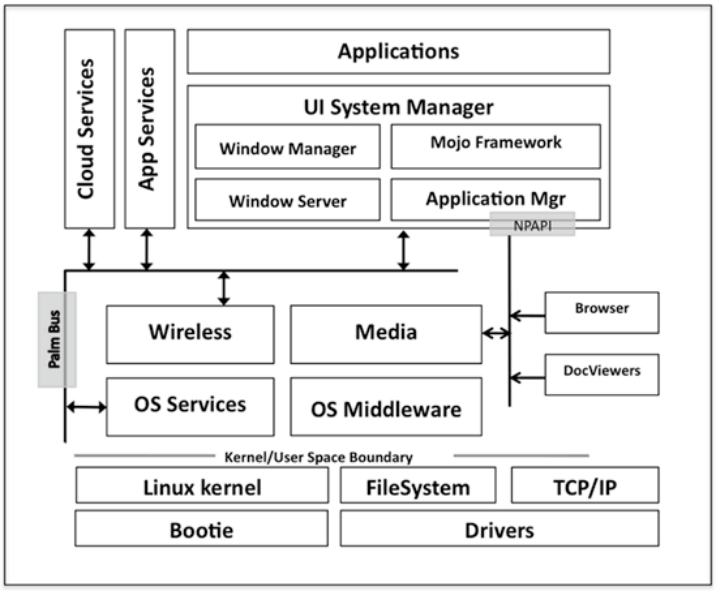

Figura 3.2: Arquitectura Palm WebOS [18]

Esta plataforma está dividida en cinco componentes, las aplicaciones del sistema, el *Framework Mojo*, el gestor UI (*User Interface*), los servicios *WebOS* y el kernel del sistema operativo. Además, los dos pilares fundamentales en la gestión de esta arquitectura son el gestor de UI y los servicios *WebOS*. En primer lugar, el gestor de UI (*UI System Manager*) es el encargado de administrar todos los recursos visuales y comunicarse con los servicios existentes del sistema operativo mediante el bus *Palm*. En segundo lugar, los servicios *WebOS* son los encargados de administrar los diferentes protocolos

de comunicación existentes, los servicios multimedia y la capa *middleware* del sistema operativo.
Finalmente, la alternativa propuesta por *Palm Inc.* plantea una serie de incógnitas sobre el futuro de los sistemas operativos móviles, tanto desde el punto de vista del usuario como desde los propios desarrolladores del sistema. Sin embargo, la actual falta de madurez proporcionada por estos sistemas, tanto a nivel de seguridad como de experiencia de usuario, retrasan el efecto esperado en el mercado de las telecomunicaciones.

3.1.4. IPHONE OS

iPhone OS es un sistema operativo monousuario y multitarea, creado recientemente por *Apple Inc.* para su principal teléfono móvil, *Apple iPhone*. Este sistema es una versión reducida de *Mac OS X* para procesadores ARM y basado en el kernel Mach. La Figura 3.3 muestra la arquitectura genérica seguida por *iPhone OS*. Esta arquitectura se encuentra compuesta por una serie de elementos que se describen a continuación [16] [55]:

- Aplicaciones:
  Esta capa contiene las aplicaciones instaladas en el teléfono móvil, y que interactúan directamente con el usuario mediante su correspondiente GUI (*Graphical User Interface*).
- Cocoa Touch:
  Esta capa contiene las interfaces necesarias para el desarrollo de aplicaciones nativas en el teléfono. Los elementos que componen esta capa son el *UIKit Framework*, *Foundation Framework* y *Address Book UI Framework*.
- Media:
  Esta capa contiene las librerías gráficas (*Quartz* y *OpenGL*), de audio (*OpenAL*) y vídeo necesarias para la utilización de este tipo de tecnologías.
- Core Services:
  Esta capa contiene la plataforma de seguridad seguida por *iPhone OS*, y otros servicios como el soporte XML (*eXtensible Markup Language*), las interfaces de geolocalización y el soporte a los protocolos de comunicación mediante el *CFNetwork*. Este componente está basado en los *BSD sockets*, y administra las pilas de los protocolos de comunicación existentes en el teléfono, mediante una capa de abstracción orientada a objetos.
- Core OS:
  Esta capa contiene las interfaces encargadas de gestionar el kernel Mach, los controladores del sistema y las funcionalidades básicas del sistema operativo.

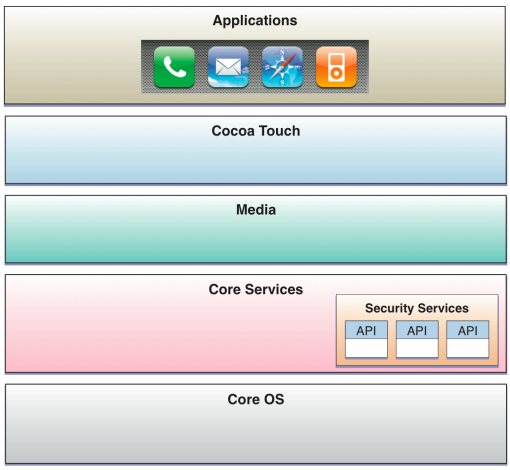

Figura 3.3: Arquitectura del iPhone OS [15]

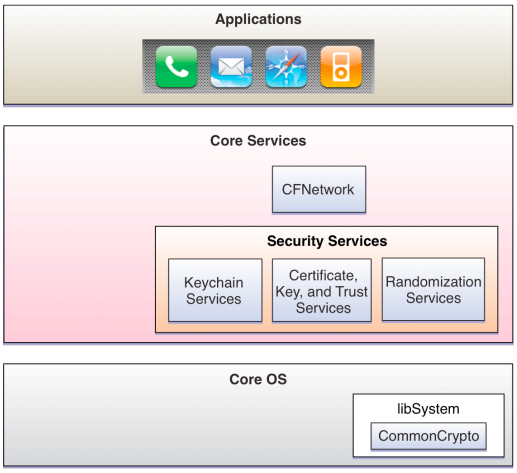

Figura 3.4: Plataforma de Seguridad [15]

La Figura 3.4 muestra la arquitectura de seguridad utilizada por *iPhone OS*. Esta plataforma contiene una serie de servicios descritos a continuación [16]:

- Servicios de Gestión de Certificados:
  Estos tipo de servicios son los responsables de garantizar la confidencialidad, la integridad y la disponibilidad de la información personal del usuario, mediante la utilización de interfaces de cifrado y gestión de certificados.
- Servicios de Aleatorización:
  Este tipo de servicios son los responsables de generar números aleatorios criptográficamente seguros.
- Servicios de Certificación:
  Este tipo de servicios son los responsables de validar los certificados proporcionados por las aplicaciones, y aplicar las políticas de seguridad correspondientes.

El flujo de trabajo seguido durante la instalación de una aplicación, valida el certificado proporcionado por la aplicación. En caso afirmativo, la aplicación se ejecutará en un entorno protegido, llamado *sandbox environment*. En caso contrario, la aplicación no podrá instalarse en el teléfono móvil.
Finalmente, la arquitectura de seguridad proporcionada por *iPhone OS* posee la política de instalación de aplicaciones más restrictiva de las plataformas analizadas. Esta característica reduce el número de vectores existentes en la plataforma. Sin embargo, la existencia de una única cuenta de usuario con permisos de administrador y el direccionamiento estático, exponen directamente la información personal del usuario una vez que un ataque ha resultado exitoso.

3.1.5. BLACKBERRY OS

Blackberry OS es un sistema operativo diseñado por RIM (*Research In Motion*) para su principal teléfono móvil orientado al mundo empresarial, *Blackberry*.

Sin embargo, este sistema monousuario y multitarea puede aplicarse en dos situaciones distintas, un entorno empresarial y un entorno doméstico. Por un lado, el entorno empresarial define una arquitectura cliente-servidor entre el teléfono móvil y el *Blackberry Enterprise Server* [17]. Este servidor proporciona una serie de servicios, que supervisan los procesos de cifrado y compresión de la información, y establecen una comunicación segura con los clientes correspondientes mediante un protocolo propietario punto a punto llamado SRP (*Server Routing Protocol*). Por último, el ámbito doméstico proporciona los casos de uso habituales en cualquier teléfono móvil, sin particularidad alguna.

3.1.6. ANDROID

Android es un sistema operativo multitarea y monousuario diseñado en primer lugar por *Google Inc.* y posteriormente por la *Open Handset Alliance*. Este sistema optimizado para procesadores ARM está basado en un *kernel* de Linux 2.6 y una máquina virtual Java optimizada para procesadores móviles, llamada *Dalvik* [24]. La Figura 3.5 muestra la organización seguida por este sistema operativo, cuya arquitectura de seguridad está basada en un esquema de seguridad a nivel de proceso, ampliamente utilizada en los sistemas Linux con UID (*User ID*) y GID (*Group ID*). Además, este sistema se caracteriza por un sistema de permisos, con varios niveles de restricción, orientado al proceso o al URI (*Uniform Resource Identifier*). En primer lugar, el sistema de protección orientado al proceso está gestionado por el instalador de paquetes, mediante el uso de certificados e interacción con el usuario. En segundo lugar, el sistema de protección orientado al URI requiere la colaboración del proveedor de contenidos para el bloqueo de los contenidos cuando sea necesario.

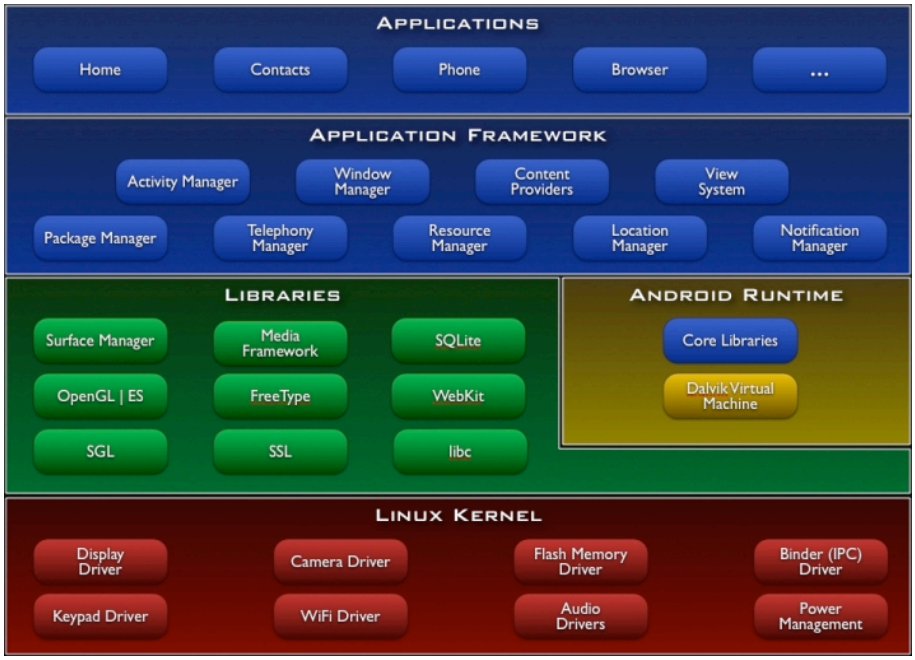

Figura 3.5: Arquitectura de Android [24]

El proceso de instalación seguido en este sistema operativo comprueba la presencia de certificados en la aplicación. En caso afirmativo, se asigna en

primer lugar los permisos correspondientes al certificado, para posteriormente asignar un identificador único UUID y ejecutar la aplicación en un entorno protegido (*sandboxed environment*). En caso contrario, la aplicación carece de permisos por defecto, y cualquier acción que ésta realice, requerirá la interacción con el usuario.
Finalmente, el sistema operativo *Android* posee una arquitectura de seguridad equilibrada, combinando las principales características de los sistemas *Symbian OS* y *Windows Mobile*. Sin embargo, la notable juventud de este sistema precisa la necesidad de un mayor tiempo de maduración que fortalezca la seguridad, especialmente en las interfaces de usuario.

3.2. VIROLOGÍA MÓVIL

Tras la presentación de los principales sistemas operativos móviles, y una breve introducción a sus plataformas de seguridad, se va a proceder al estudio de la virología móvil. Además, este tipo de virología posee una infraestructura compuesta por la morfología y la taxonomía vírica, y que son a su vez utilizados en la virología aplicada a los ordenadores personales. Sin embargo, antes de comenzar el siguiente estudio es necesario definir las principales familias de virus que componen el panorama actual [20]:

- Virus basados en la propagación o *Cabir*.
  Este tipo de virus utiliza los protocolos de comunicación más conocidos como Bluetooth y *Wi-Fi* [1] para replicarse de un teléfono móvil a otro, sin ocasionar daño económico al usuario ni capturar o alterar su información personal.
- Virus basados en la modificación del sistema operativo o *Skulls*.
  Este tipo de virus modifica los archivos del sistema operativo con el objetivo de inutilizar funcionalidades específicas o cambiar el aspecto de la interfaz de usuario.
- Virus basados en los daños económicos o *Commwar*.
  Este tipo de virus utiliza la información personal del usuario, como por ejemplo su lista de contactos, para propagarse a través de MMS, Bluetooth [35] o *Wi-Fi* [1], y ocasionando un daño económico al usuario.

3.2.1. MORFOLOGÍA VÍRICA

La morfología vírica se define como la parte de la virología móvil que ocupa la estructura de los virus, y dividida a su vez en dos bloques fundamentales, la morfología interna y externa. En primer lugar, la morfología externa define el comportamiento de la aplicación maliciosa dentro del teléfono móvil infectado. En segundo lugar, la morfología interna describe los detalles de implementación que caracterizan su desarrollo.
La Figura 3.6 muestra la estructura principal de los virus en la telefonía móvil, que fundamenta los sistemas de clasificación actuales.

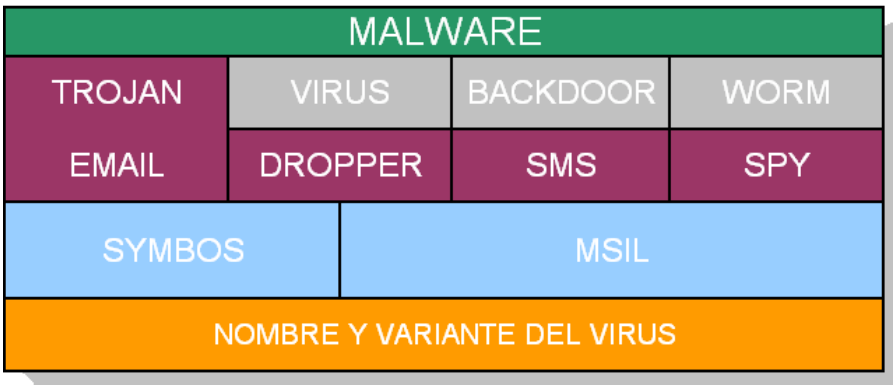

Figura 3.6: Diagrama de la Morfología Vírica en la Telefonía Móvil

Este modelo organizativo contiene los elementos principales que conforman el nombre de los virus para dispositivos móviles. En primer lugar, se debe indicar el tipo de *malware* que corresponde con el virus concreto. En segundo, se debe indicar el sistema operativo afectado por el virus seleccionado. Finalmente, se especifica una variante distinta si el comportamiento o el código fuente del virus es diferente respecto a sus antecesores.
Además, este modelo basado en capas posee una serie de ventajas e inconvenientes que se detallan a continuación:

- Facilidad de Visualización:
  Esta estructura favorece la visualización de los virus dentro de las listas de clasificación de los antivirus actuales.
- Escalabilidad:
  Esta organización favorece la escalabilidad de la arquitectura, permitiendo la inclusión de nuevos módulos sin alterar el comportamiento global del sistema de clasificación.

Como se comentó en la introducción de la sección, la morfología externa define el comportamiento seguido por los virus tras su instalación en el teléfono móvil infectado. Sin embargo, este modo de funcionamiento sigue unos patrones claramente definidos, que permiten su clasificación mediante el uso de parámetros como la familia del virus, el protocolo de propagación, la variante y el esquema de propagación. La familia del virus identifica la tipología o el comportamiento externo seguido por el correspondiente virus. El protocolo y esquema de propagación definen el medio de comunicación utilizado para su propagación, y el árbol de directorio del virus respectivamente. Por último, la variante del virus determina su variación respecto al modelo original que identifica la familia del virus. El siguiente ejemplo muestra la morfología externa de un virus de propagación (*Cabir*) mediante la utilización de este modelo de clasificación de virus, utilizado por conocidas compañías de antivirus como *F-Secure* [21].

- **Nombre:** *Cabir*
- **Familia de Virus:** Propagación
- **Protocolo de Propagación:** Bluetooth
- **Variante:** B.
- **Esquema de Propagación:**

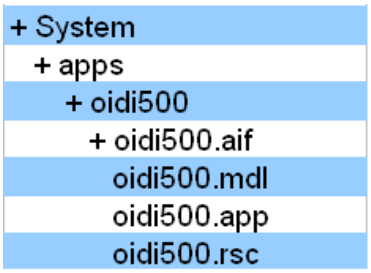

Figura 3.7: Esquema de Propagación de *Cabir*

Por último, la morfología interna define los detalles de implementación necesarios para la creación de un virus en una plataforma móvil. Además, los puntos clave que determinan este tipo de morfología son el lenguaje de programación y el protocolo de comunicación utilizado. Debido al problema de la fragmentación móvil, cada lenguaje de programación se encuentra fuertemente unido a un sistema operativo concreto, lo que limita pero no elimina, las posibilidades de propagación entre plataformas. La Tabla 3.1 muestra la asociación correspondiente entre el sistema operativo móvil y su principal lenguaje de programación.

Sistema Operativo	Lenguaje de Programación
Symbian OS	C++
Windows Mobile CE	C#
Google Android	J2ME
Blackberry	J2ME
Palm WebOS	XHTML, JavaScript, CSS
iPhone OS	Objective-C, C++ (Cocoa)

Tabla 3.1: Relación entre OS y lenguajes de programación

El protocolo de comunicación es uno de los factores principales que influyen en la morfología interna de un virus. Sin embargo, este componente depende directamente de las interfaces de comunicación que proporcione el sistema operativo mediante su lenguaje de propagación, por lo que cada caso de propagación varía en función del sistema operativo utilizado.

3.2.2. TAXONOMÍA VÍRICA

La taxonomía se define como la parte de la virología móvil que define los principios, métodos y fines de clasificación de virus. Además, este concepto es ampliamente utilizado por los sistemas de antivirus para detectar nuevas, o existentes amenazas para los usuarios. Por consiguiente, el siguiente apartado describe cronológicamente los diferentes sistemas de clasificación.
Actualmente, uno de los principales retos de las compañías de antivirus es la creación y mantenimiento de un sistema de clasificación robusto que, permita determinar la familia del nuevo virus con cierta eficacia y eficiencia. Sin embargo, los sistemas de clasificación han evolucionado respecto al incremento del número de familias de virus. En otoño de 2004, apareció la primera clasificación mediante las siguientes tres tendencias [20]:

- Amenazas basadas en pérdidas financieras o *Adware*.
- Amenazas basadas en la modificación del sistema operativo.
- Amenazas basadas en la propagación.

Además, este aumento de las familias de virus incrementó la necesidad de crear mejores sistemas de clasificación que mejoraran la metodología existente. Ejemplos de esta evolución se presentaron mediante sistemas de clasificación basados en los siguientes principios:

- Conducta o comportamiento del virus.
- Entorno de ejecución.
- Nombre de la familia y letra de la variante.

Sin embargo, a pesar de esta evolución, no se llegó a resolver de manera eficaz el problema de la hibridación, basado en la mezcla entre familias de virus [20]. La problemática surgida por la hibridación incrementó el número de familias de virus ya existentes, dificultando así el proceso de documentación y tratamiento de cada nueva variante de un virus.

Figura 3.8: Esquema de taxonomía vírica [22]

La Figura 3.8 muestra un modelo de representación más detallado, que cubre la mayoría de las características necesarias durante ataques utilizados en las virologías existentes de la seguridad informática actual. El presente gráfico está compuesto por cinco tipos de características, cada una con sus correspondientes variantes, descubrimiento de dispositivos, transmisión al dispositivo, activación, *payloads* y motivación del atacante. El principal objetivo de este nuevo esquema de representación y clasificación es la reducción del impacto producido por factores como la fragmentación móvil y la hibridación.

Actualmente, organizaciones no lucrativas, constituidas por comités de expertos e investigadores en seguridad informática, se encuentran implementando metodologías de buenas prácticas con el objetivo de fortalecer los modelos ya existentes. En primer lugar, la organización CARO (*Computer Antivirus Researcher's Organization*) presenta un nuevo esquema de clasificación basado en los siguientes puntos:

- Familia
- Grupo
- Variante *Major*
- Variante *Minor*
- Modificador

En segundo y último lugar, la organización OMTP (*Open Mobile Terminal Platform*) diseñó un esquema análogo representado mediante la siguiente convención [23]:

- Amenazas de modificación de software (T.SWM.xxx).
- Amenazas de oportunismo del software (T.SWO.xxx).
- Amenazas Hardware Externas (T.HWE.xxx).
- Amenazas Hardware de intrusión del terminal (T.HWT.xxx).
- Amenazas Hardware de componentes invasivos (T.HWC.xxx).
- Amenazas Hardware de suplantación (T.CLO.xxx).

3.3. CONCLUSIONES

El presente capítulo ha consistido en el estudio del estado del arte de la virología móvil. Este estudio estaba compuesto por una visión práctica, mediante la enumeración y descripción de los principales sistemas operativos móviles, y una versión teórica mediante la definición de los conceptos de morfología y taxonomía de los modelos de virus para dispositivos móviles. De esta manera, gracias al presente estudio del estado del arte, es posible reconocer las principales amenazas y riesgos de las plataformas de seguridad de cada sistema operativo móvil, y mediante el diseño del sistema propuesto contrarrestar estos efectos negativos.
El siguiente capítulo finaliza el apartado introductorio de los estudios del estado del arte, con la descripción del protocolo de comunicaciones Bluetooth.

4. SISTEMAS DE COMUNICACIÓN BLUETOOTH

La actual evolución de los dispositivos móviles ha venido determinada por el creciente número de funcionalidades requeridas por los usuarios. Sin embargo, este proceso evolutivo ha repercutido en otros protocolos de comunicación, como por ejemplo Bluetooth [35]. Sus orígenes datan de 1994, cuando dos desarrolladores de *Ericsson Mobile Platforms*, Jaap Haartsen y Sven Mattisson, desarrollaron la primera especificación Bluetooth, basada en la tecnología FHSS (*Frequency-Hopping Spread Spectrum*), que fue rápidamente formalizada y estandarizada por el Bluetooth SIG (*Special Interest Group*) [36] hasta llegar a su actual versión 3.0. El siguiente estado del arte tiene por objeto el estudio de la especificación Bluetooth, mediante el análisis de sus dos volúmenes:

- Volumen 1: *Core*.
- Volumen 2: *Profiles*.

Por este motivo, la creciente utilización del protocolo de comunicaciones Bluetooth, junto con el aumento de sus funcionalidades, y la incorporación en la mayoría de dispositivos móviles, lo ha convertido en uno de los sistemas con mayor porvenir en las comunicaciones inalámbricas móviles.

Por último, el presente capítulo, consistente en la descripción de la especificación Bluetooth, está compuesta, principalmente, por dos secciones principales o volúmenes. En primer lugar, la sección *core* contiene los elementos que constituyen las bases del protocolo, tanto a nivel eléctrico como de telecomunicaciones. En segundo lugar, la sección *profiles* contiene las funcionalidades del protocolo utilizadas, principalmente, a nivel de aplicación.

4.1. VOLUMEN 1: CORE

El volumen 1 de la especificación Bluetooth describe el núcleo del protocolo mediante la descripción de los detalles de más bajo nivel, y que incluye los siguientes apartados:

- Arquitectura de la especificación
- Niveles físicos de la especificación
    - Especificación de radiofrecuencia.
    - Especificación de los protocolos de banda base y rutinas de enlace de bajo nivel.
    - Especificación LMP (*Link Manager Protocol*).
    - Especificación HCI (*Host Controller Interface*).
    - Especificación de los códigos de error y paso de mensajes a nivel de enlace.
    - Especificación de seguridad a nivel de enlace.

- Nivel de enlace de la especificación
  - Especificación L2CAP (*Logical Link Control Adaptation Protocol*).
  - Especificación SDP (*Service Discovery Protocol*).
  - Especificación GAP (*Generic Access Profile*).
- Nivel de transporte de la especificación
  - Especificación de la capa de transporte UART.
  - Especificación de la capa de transporte USB.
  - Especificación de la capa de transporte SD.
  - Especificación de la capa de transporte *Three-wire* UART.

4.1.1. ESPECIFICACIÓN DE RADIOFRECUENCIA

El protocolo de comunicación Bluetooth trabaja en la banda libre de frecuencias ISM 2,4 GHz mediante la tecnología FHSS (*Frequency-Hopping Spread Spectrum*). Esta técnica, basada en saltos en frecuencia, define en el caso general, 79 saltos de 1 MHz en la banda de frecuencias entre 2402 y 2480 MHz, permitiendo así realizar 1600 saltos por segundo. Además, la técnica de modulación de la señal que emplea es GFSK (*Gaussian Frequency Shift Keying*). Tras describir el modo físico de funcionamiento, cabe destacar que la velocidad máxima de transmisión depende directamente del núcleo utilizado, y que la potencia de transmisión se estructura en tres clases diferentes de productos [36]:

- Clase 1: 100 mW / 20 dBm con un alcance aproximado de 100 metros.
- Clase 2: 2.5 mW / 4 dBm con un alcance aproximado de 10 metros.
- Clase 3: 1 mW / 0 dBm con un alcance aproximado de 1 metro.

Asimismo, la potencia de transmisión define dos modos principales de funcionamiento [38]:

- Modos de alto consumo de potencia
  - *Discoverable mode.*
  - *Connectable mode.*
- Modos de bajo consumo de potencia
  - *Hold mode.*
  - *Sniff mode.*
  - *Park mode.*

4.1.2. ESPECIFICACIÓN DE LOS PROTOCOLOS DE BANDA BASE

Los protocolos de comunicación en banda base definen diferentes canales lógicos a partir de las secuencias de saltos establecidas en las distintas bandas de frecuencias, siendo posteriormente multiplexadas mediante técnicas basadas en división de tiempo TDD (*Time Division Multiplexing*). El principal objetivo de estos protocolos consiste en el establecimiento del enlace físico por radiofrecuencia mediante técnicas de modulación y demodulación de la señal. Por último, los enlaces físicos pueden ser de dos tipos [37]:

- SCO (*Synchronous Connection-Oriented*)

Conexiones simétricas punto a punto capaz de soportar tráfico en tiempo real.
- ACL (*Asynchronous Connection-Less*)
  Conexiones simétricas o asimétricas punto a multipunto capaz de soportar altas tasas de tráfico de datos.

4.1.3. ESPECIFICACIÓN LMP

El protocolo de control de enlace LMP es el encargado de inicializar, configurar y monitorizar el enlace entre dos dispositivos. Además, este nivel de comunicación está implementado en el propio controlador y define una serie de PDU (*Process Data Unit*), con mayor preferencia sobre los datos del usuario, para un intercambio de la información sin retardos.

4.1.4. ESPECIFICACIÓN HCI

El protocolo HCI es el encargado de establecer una comunicación, estandarizada y de bajo coste de integración, entre el controlador físico y la pila de comunicaciones Bluetooth del dispositivo. Además, proporciona una interfaz de acceso a las funciones de niveles inferiores mediante los siguientes elementos:
- *HCI Driver.*
- *HCI Firmware.*
- *Host Controller Transport Layer.*

4.1.5. ESPECIFICACIÓN DE SEGURIDAD

La especificación de seguridad define los principales componentes de seguridad a nivel de enlace que garantizan la privacidad, confidencialidad e integridad de la información transmitida [39]:
- Gestión de claves.
- Técnicas de cifrado.
- Técnicas de autenticación.

En primer lugar, la gestión de claves viene determinada por el esquema de la Figura 4.1, donde cada una de las claves descritas posee una funcionalidad concreta dentro de su contexto, tanto de autenticación como de cifrado.

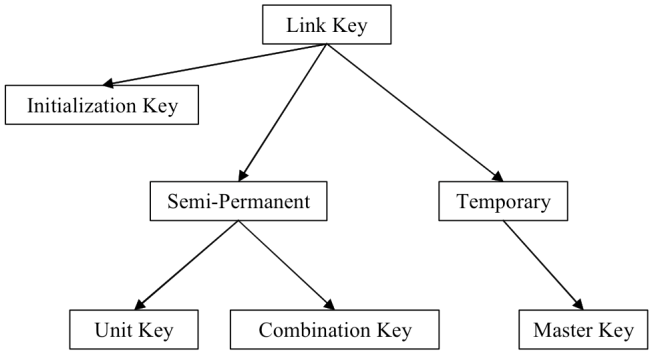

Figura 4.1: Esquema de claves de enlace Bluetooth [39]

En segundo lugar, las técnicas de cifrado determinan los cifrados de flujo basados en el algoritmo E_0 y los diferentes modos de seguridad en función del tipo de clave de enlace empleada. Por este motivo, las claves de enlace semi-permanentes no cifran el tráfico en difusión y opcionalmente cifran el tráfico individual. Sin embargo, las claves temporales definen tres modos de seguridad a nivel de enlace [39]:

- Modo 1: Sin seguridad.
- Modo 2: Tráfico en difusión sin cifrar y tráfico individual cifrado.
- Modo 3: Tráfico en difusión e individual cifrado.

Por último, las técnicas de autenticación están basadas en algoritmos de desafío-respuesta mediante el uso de claves simétricas privadas y tiempos de espera entre autenticaciones.

4.1.6. ESPECIFICACIÓN L2CAP

El protocolo de control y adaptación de nivel de enlace L2CAP es el encargado de proporcionar técnicas de multiplexación, segmentación y reensamblado de paquetes, que garanticen la interoperabilidad entre los niveles físicos y los niveles de transporte y aplicación. Además, este protocolo se utiliza exclusivamente en conexiones ACL y establece los parámetros de calidad de servicio *QoS* entre los dispositivos.

4.1.7. ESPECIFICACIÓN SDP

El protocolo de búsqueda de servicios SDP es el encargado, mediante mecanismos de desafío-respuesta, de obtener los servicios de cada dispositivo detectado. Los servicios encontrados tienen correspondencia directa con un perfil Bluetooth concreto y están compuestos por los siguientes elementos [37]:

- GUID (*Globally Unique Identifier*)
  Identificador único que representa un perfil concreto.
- *Service Class Instance*
  Instancia que representa los servicios genéricos soportados y el tipo de dispositivo.
- *Service Record Instance*
  Instancia que representa las características principales del perfil.

4.2. VOLUMEN 2: PROFILES

El volumen 2 de la especificación Bluetooth define el concepto de perfil de comunicaciones como la especificación formalizada de una interfaz de alto nivel que permite la interconexión entre dos dispositivos en relación maestro-esclavo. Por consiguiente, el aseguramiento de la interoperabilidad entre ambos dispositivos requiere que, tanto maestro como esclavo, posean el perfil de comunicación correspondiente. En primer lugar, la especificación Bluetooth define cuatro perfiles genéricos:

- *Generic Access Profile (GAP).* Perfil de acceso genérico.

- *Serial Port Protocol (SPP).* Perfil de comunicación a través de una emulación del puerto serie.
- *Service Discovery Application Profile (SDAP).* Perfil de búsqueda de aplicaciones.
- *Generic Object Exchange Profile (GOEP).* Perfil genérico de intercambio de archivos.
- *Generic Audio/Video Distribution Profile (GAVDP).* Perfil generico de *streaming* de audio o video entre dos dispositivos.

Una vez descritos los perfiles de primer nivel se definen los perfiles secundarios más conocidos de la especificación Bluetooth [36] y su representación genérica mediante la Figura 20:

- *Advanced Audio Distribution Profile (A2DP).* Perfil de *streaming* de audio (mono o estéreo) entre dos dispositivos basado en el perfil GAVDP (*Generic Audio/Video Distribution Profile*).
- *Audio/Video Remote Control Profile (AVRCP).* Perfil de control remoto de dispositivos de audio y video basado en el perfil SPP (*Serial Port Profile*).
- *Basic Imaging Profile (BIP).* Perfil de tratamiento de imágenes (transporte, control remoto e impresión).
- *Basic Printing Profile (BPP).* Perfil de impresión (utilización básico y control remoto).
- *Video Distribution Profile (VDP).* Perfil de *streaming* de video entre dos dispositivos basado en el perfil GAVDP (*Generic Audio/Video Distribution Profile*).
- *irMC Synchronization.* Perfil de sincronización e intercambio de ficheros entre dos dispositivos.
- *SIM Access Profile (SAP).* Perfil de acceso a la tarjeta SIM (*Subscriber Identity Module*) del dispositivo maestro.
- *Phone Book Access Profile (PBAP).* Perfil de acceso a la agenda de teléfonos y su envío entre dos dispostivos.
- *Personal Area Networking (PAN).* Perfil que permite la encapsulación de Bluetooth sobre protocolos de nivel de red, como por ejemplo IP (*Internet Protocol*).
- *Object Push Profile (OPP).* Perfil de envío basado en el perfil GOEP (*Generic Object Exchange Protocol*) básico de objetos genéricos entre dos dispositivos.
- *Dial-up Networking (DUN).* Perfil de comunicación serie basado en el perfil SPP (*Serial Port Profile*) entre dos dispositivos mediante el uso de comandos AT.
- *Objext Exchange File Transfer Profile (OBEX FTP).* Perfil de transferencia y exploración de ficheros basado en el perfil GOEP (*Generic Object Exchange Protocol*) entre dos dispositivos.
- *Headset Profile (HSP).* Perfil de *streaming* de audio mediante auriculares basado en el perfil SPP (*Serial Port Profile*).
- *Hands-free Profile (HFP).* Perfil de *streaming* de audio mediante el uso del altavoz del dispositivo móvil.

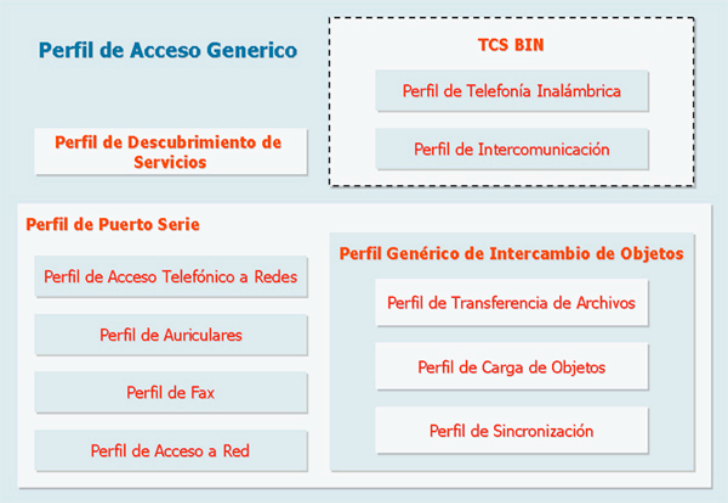

Figura 4.2: Esquema de perfiles Bluetooth [37]

Por último, cada perfil de comunicación define su propio protocolo de comunicación y el formato de los mensajes que transmite para implementar una funcionalidad concreta. Sin embargo, la presencia obligatoria u opcional de estas funcionalidades depende estrictamente de la propia especificación Bluetooth.

4.3. CONCLUSIONES

El presente capítulo ha consistido en el estudio del estado del arte del protocolo de comunicaciones Bluetooth. Esta introducción teórica se ha basado en la definición de los principales bloques que componen la especificación, con el objetivo de estudiar su funcionamiento y poder aplicarlo al diseño del sistema propuesto. La modularidad y escalabilidad de este protocolo, mediante sus diferentes *profiles*, han sido factores discriminantes a la hora de elegir el protocolo de comunicaciones, e incluso, han facilitado la implementación del sistema propuesto.
El siguiente capítulo ha procedido al análisis del sistema propuesto, mediante la especificación de sus principales requisitos, y que han sido utilizados a posteriori por las fases diseño e implementación.

5. ANÁLISIS DEL PROYECTO

La fase de análisis del proyecto, definida por el documento de especificación de requisitos, tiene por objeto introducir las principales características del sistema de detección de intrusiones Bluetooth para dispositivos móviles, y que posteriormente han sido utilizados como referencia en las etapas de diseño e implementación. Además, han sido evaluados la complejidad de implementación del sistema dentro de una plataforma de desarrollo concreta. Este conjunto de funcionalidades está estructurado en cuatro bloques principales:

- Arquitectura modular distribuida.
- Protocolo de comunicación.
- Especificación de la plataforma de desarrollo.
- Modos de operación del sistema.
- Aplicaciones futuras.

El presente capítulo tiene por objeto especificar los principales requisitos del sistema propuesto, que han sido posteriormente utilizados por las fases de diseño e implementación. Además, este capítulo está compuesto por la especificación de la arquitectura principal del sistema propuesto, el proceso de selección y especificación del protocolo de comunicaciones Bluetooth, el proceso de selección y especificación de la plataforma de desarrollo *Windows Mobile*, la enumeración y descripción de los tres modos de operación existentes en el sistema, y sus correspondientes aplicaciones futuras.

5.1. ARQUITECTURA MODULAR HÍBRIDA

El diseño de una arquitectura modular está basado en el estándar creado por los grupos de trabajo CIDF [27] e IDWG dentro del estudio del estado del arte sobre los sistemas de detección de intrusiones. Este modelo basado en componentes constaba de cuatro bloques:

- Bloques encargados de obtener información del sistema.
- Bloques encargados de analizar la información del sistema.
- Bloques encargados de almacenar y proporcionar información complementaria al análisis.
- Bloques encargados de actuar en función de la respuesta generada durante el proceso de análisis.

Sin embargo, esta arquitectura puede enfocarse desde diferentes puntos de vista, un modelo de procesamiento centralizado o distribuido. En primer lugar, el enfoque centralizado aporta una mayor facilidad de implementación y un alto coste computacional en el procesamiento y almacenamiento de la información. Por el contrario, el enfoque distribuido aporta una mayor complejidad de implementación pero una mejor carga computacional y de almacenamiento del sistema. Por consiguiente, el enfoque distribuido ofrece un mejor rendimiento, respecto al modelo centralizado, dentro de un entorno

móvil, donde el coste computacional es un factor crítico de diseño. La Tabla 5.1 muestra un resumen con las principales características entre ambos puntos de vista.

Modelo Centralizado	Modelo distribuido
– Facilidad de implementación. – Alto coste computacional. – Alta carga de almacenamiento. – Baja tolerancia a fallos. – Complejidad de mantenimiento.	– Complejidad de implementación. – Bajo coste computacional. – Baja carga de almacenamiento. – Alta tolerancia a fallos. – Facilidad de mantenimiento.

Tabla 5.1: Comparación entre modelos de IDS

Por este motivo, la necesidad de diseñar e implementar un sistema basado en una arquitectura distribuida está fundamentada, principalmente, en las limitaciones computacionales de los dispositivos móviles actuales.

5.2. PROTOCOLO DE COMUNICACIÓN

El diseño del protocolo de comunicación está basado, al igual que en el apartado anterior, en los estándares de intercambio de mensajes IDXP, y formato de la información a transmitir IDMEF, definidos en el estudio del estado del arte de los sistemas de detección de intrusiones. Por consiguiente, este modelo de comunicación tiene por objeto garantizar la interoperabilidad entre los diferentes módulos que componen el sistema, con independencia de su localización geográfica, debido a su arquitectura distribuida, y proporcionar así un nivel de abstracción que facilite las tareas de implementación. Además, otro de los principales factores a tener en cuenta a la hora de definir el protocolo de comunicación es el medio a través del cual se transmitirá la información. Actualmente, los principales medios de transmisión de datos existentes en la telefonía móvil son:

- Bluetooth [35].
- *Wi-Fi* [1].
- Protocolos de acceso de paquetes a redes telefónicas
  - GPRS (*General Packet Radio Service*).
  - UMTS (*Universal Mobile Telecommunications System*).
  - HSPA (*High-Speed Packet Access*).

Sin embargo, estos medios no siempre estarán accesibles o habilitados para el intercambio de información, siendo necesario establecer diferentes niveles de prioridades que garanticen una alta disponibilidad del servicio. Por este motivo, los protocolos de acceso de paquetes a redes telefónicas son el medio de transmisión con mayor prioridad debido a su alta disponibilidad. Posteriormente, se utilizará Bluetooth o *WiFi* dependiendo de la distancia entre los módulos del sistema y la existencia de puntos de acceso que aseguren una conectividad apropiada. Por último, la Tabla 5.2 muestra un

resumen de las principales características de los medios de transmisión descritos.

Bluetooth	Wi-Fi	WiMAX	GPRS, UMTS, HSPA
– Bajo alcance. – Sin necesidad de puntos de acceso. – Sin tarificación.	– Medio alcance. – Necesidad de puntos de acceso. – Tarificación opcional.	– Alto alcance. – Necesidad de puntos de acceso. – Tarificación .	– Alto alcance. – Necesidad de puntos de acceso. – Tarificación obligatoria.

Tabla 5.2: Comparación de medios de transmisión

Finalmente, la elección del protocolo de comunicaciones Bluetooth se ha debido a su creciente utilización, la relevancia en el desarrollo de aplicaciones de seguridad, y la independencia respecto al operador de red (ISP).

5.3. ESPECIFICACIÓN DE LA PLATAFORMA DE DESARROLLO

El siguiente proceso de especificación está basado en las principales características de los sistemas operativos móviles estudiados en el estado del arte de la virología móvil, y su descomposición en los siguientes factores:

- Tipo de SDK (*Software Development Kit*).
- Dificultad de implementación.
- Cuota de mercado.
- Obtención de terminales de prueba.

En lo referente al tipo de entorno de desarrollo o SDK, la mayoría de los sistemas operativos estudiados proporcionan entornos abiertos y gratuitos, con algunas reducidas excepciones. La dificultad de implementación define tanto la complejidad del lenguaje de programación, como las limitaciones proporcionadas por las interfaces del sistema operativo. Por este motivo, la mayoría de los sistemas evaluados proporcionan interfaces bastante limitadas para la implementación de aplicaciones basadas en Bluetooth [35].

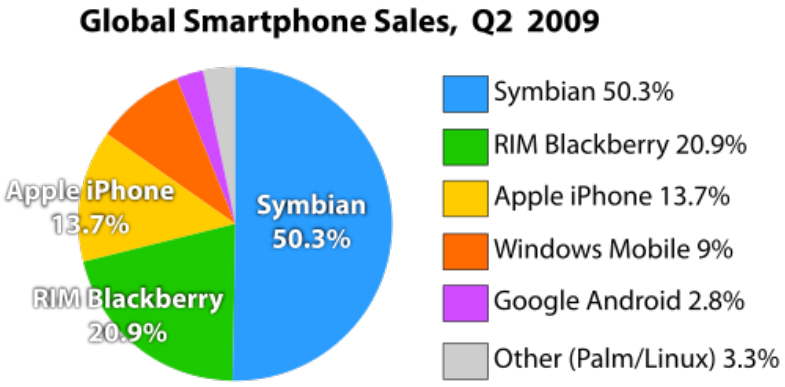

Figura 5.1: Cuota de Mercado de OS móviles [53]

La Figura 5.1 muestra la cuota de mercado de los sistemas operativos más utilizados y su correspondiente relevancia en el mercado de las

telecomunicaciones. En general, las plataformas más utilizadas son aquellas con mayor madurez, como por ejemplo *Symbian OS*, *RIM* y *Windows Mobile*. Sin embargo, la reciente aparición de plataformas como *Android*, y especialmente *iPhone OS*, está modificando la cuota de mercado existente a favor de estos nuevos terminales con mayores y mejores prestaciones para los usuarios. Finalmente, la obtención de terminales de prueba es uno de los elementos esenciales para la validación del propio sistema, ya que sin éstos no se podría dar una utilidad real a la aplicación. Por este motivo, el siguiente factor afecta en mayor medida a las plataformas más recientes, tales como *Android* y *iPhone OS*, ya que poseen una menor gama de terminales en el mercado. La Tabla 5.3 muestra un resumen de las principales características de los sistemas operativos analizados.

Symbian OS	Windows Mobile	Android	iPhone OS
– Sistema maduro. – SDK gratuito. – Muchos terminales. – C++.	– Sistema maduro. – SDK gratuito. – Muchos terminales. – C#/C++.	– Sistema reciente. – SDK gratuito. – Pocos terminales. – Java.	– Sistema reciente. – SDK de pago. – Pocos terminales. – Cocoa.

Tabla 5.3: Comparativa de sistemas operativos móviles

Finalmente, el sistema operativo seleccionado como plataforma de desarrollo ha sido *Windows Mobile*, por la facilidad y extensibilidad del lenguaje de programación *C#* dentro del entorno de trabajo .NET, la extensa y completa documentación de las interfaces de desarrollo mediante MSDN (*Microsoft Developer Network*), y su significativa cuota de mercado. Sin embargo, las notables limitaciones en las interfaces para el desarrollo de aplicaciones basadas en Bluetooth han sido tratadas con mayor detalle a lo largo de la fase de diseño.

5.4. MODOS DE OPERACIÓN

Los modos de funcionamiento previstos para el sistema se estructuran en dos bloques principales:

- Modo de funcionamiento estándar.
- Modo de funcionamiento adaptativo.

Sin embargo, la explicación de ambos modos requiere la introducción de los perfiles Bluetooth, definidos durante el estudio del estado del arte de los sistemas de comunicación Bluetooth. La especificación de estos perfiles viene definida en el Volumen 2 de su arquitectura, y determina los diferentes servicios y funcionalidades que ofrece un dispositivo móvil mediante el correspondiente protocolo de comunicación. Asimismo, la principal diferencia entre ambos radica en los perfiles a monitorizar, ya que el procesamiento de la información es idéntico para ambos modos. En primer lugar, el funcionamiento estándar ha monitorizado todos los perfiles existentes en la *stack* Bluetooth del dispositivo móvil, para su posterior análisis y generación de una respuesta adecuada a cada caso. Por el contrario, el funcionamiento

adaptativo ha monitorizado los perfiles detectados mediante búsquedas periódicas de dispositivos próximos mediante el perfil SDP (*Service Discovery Protocol*). La periodicidad de las búsquedas vendrá determinada por el retardo fijado por el sistema operativo y un factor adicional que evite una sobrecarga del sistema. Finalmente, el modo de funcionamiento adaptativo optimiza el rendimiento del sistema, pero su lógica computacional es más compleja respecto al funcionamiento estándar.

5.5. APLICACIONES FUTURAS

La descripción de los principales modos de funcionamiento ha definido el comportamiento estándar del sistema de detección de intrusiones. Sin embargo, este modelo de arquitectura distribuida puede ser utilizado en múltiples entornos móviles, tanto empresariales como domésticos. Por consiguiente, la utilización de esta arquitectura en entornos de red basados en políticas, como por ejemplo PEP (*Policy Enforcement Point*), PDP (*Policy Decision Point*), o controles parentales, ofrecen un enfoque de control y monitorización de las acciones realizadas sobre un dispositivo móvil en entornos empresariales y domésticos respectivamente. Además, este proceso de monitorización ha permitido garantizar la privacidad de los datos, tanto empresariales como personales, y la confidencialidad de las comunicaciones establecidas.

PEP/PDP		Control Parental	
Ventajas	Desventajas	Ventajas	Desventajas
– Privacidad de la información empresarial. – Confidencialidad de las comunicaciones.	– Consumo computacional.	– Privacidad de la información personal. – Confidencialidad en las comunicaciones.	– Consumo computacional.

Tabla 5.4: Comparativa de aplicaciones futuras

La Tabla 5.4 muestra una breve comparativa con las principales ventajas e inconvenientes generadas por estas aplicaciones derivadas del sistema principal.

5.6. CONCLUSIONES

El presente capítulo ha introducido los principales requisitos del sistema, con el objetivo de permitir a las fases posteriores de diseño e implementación el desarrollo de los puntos aquí citados. La especificación de una arquitectura modular híbrida ha permitido un proceso de monitorización amplio y eficiente. La selección y descripción del protocolo de comunicaciones Bluetooth se ha basado en la comparativa respecto a protocolos de comunicación inalámbricos como *GPRS*, *UMTS* o *Wi-Fi*. De igual manera, la selección y descripción de la plataforma de desarrollo *Windows Mobile* se ha basado en función de parámetros como la facilidad de implementación o la disposición de terminales de prueba. Finalmente, los distintos modos de operación del sistema propuesto, y sus correspondientes aplicaciones futuras, son variaciones funcionales de la arquitectura principal.
El siguiente capítulo ha procedido a la descripción del diseño del sistema, mediante la definición de los detalles y limitaciones de implementación de la plataforma de desarrollo, y el protocolo de comunicaciones seleccionado.

6. DISEÑO DEL PROYECTO

Tras la fase de análisis, se han enumerado los principales requisitos funcionales que caracterizarán el sistema final. La selección de *Windows Mobile* como plataforma de desarrollo y Bluetooth como protocolo de comunicaciones a analizar, junto con la especificación de la arquitectura modular del sistema, permiten evaluar la complejidad del diseño e implementación del prototipo final. Por este motivo, la fase de diseño tiene por objeto describir la funcionalidad del sistema, junto con los principales detalles de implementación que serán necesarios durante la fase de desarrollo del proyecto. El estudio de los estados del arte de capítulos anteriores, junto con el conocimiento previo de programación en la plataforma .NET, son necesarios para la compresión del proceso de implementación.
Por último, esta fase está compuesta por tres bloques principales:

- Sistema de detección de intrusiones para terminales móviles.
- Modelo de detección adaptativo sobre Bluetooth.
- Sistema basado en políticas PEP/PDP (Policy Enforcement Point / Policy Decision Point) para terminales móviles.

6.1. SISTEMA DE DETECCIÓN DE INTRUSIONES PARA TERMINALES MÓVILES

Todo diseño de un sistema de detección de intrusiones debe comenzar por la descripción de su arquitectura. De esta manera, se enumeran los principales componentes que lo constituyen, y se descompone el problema en pequeños apartados con el objetivo de facilitar las tareas de implementación.

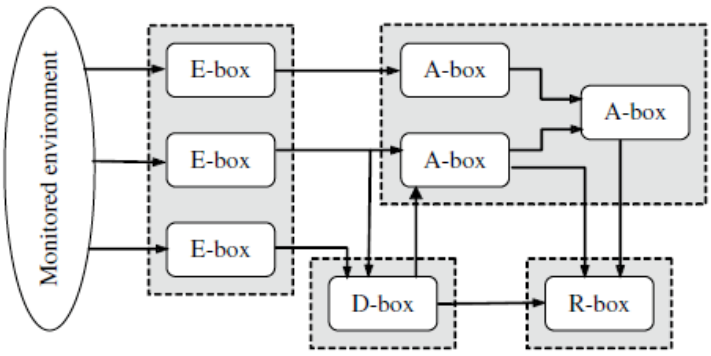

Figura 6.1: Arquitectura CIDF [27]

La Figura 6.1 muestra el diseño de la arquitectura seguida. Esta arquitectura está basada en el modelo descrito por el grupo de trabajo CIDF [27], que descompone el sistema global en cuatro componentes principales:

- Bloque E

Este bloque es el encargado de monitorizar y capturar las señales que genere el sistema.

- Bloque A
  Este bloque es el encargado de analizar las señales capturadas por el bloque E y determinar su peligrosidad.
- Bloque D
  Este bloque es el encargado de almacenar la información relativa con las posibles amenazas que puedan afectar al comportamiento del sistema.
- Bloque R
  Este bloque es el encargado de analizar la respuesta generada por el bloque A y realizar una acción concreta sobre el sistema.

El modelo propuesto presenta una arquitectura centralizada donde todos los componentes se encuentran, generalmente, en el mismo punto o sistema. Por el contrario, la mayoría de los terminales móviles no pueden aceptar este modelo de representación debido a las actuales limitaciones computaciones del hardware. Por consiguiente, tanto el coste computacional como el nivel de almacenamiento de este modelo penalizaría el rendimiento del sistema, provocando a su vez una mala experiencia de usuario. Asimismo, el diseño de una arquitectura distribuida reduciría la carga computacional del sistema, permitiendo así, la instalación y ejecución de este tipo de sistemas en un terminal móvil.
El modelo distribuido está compuesto por dos puntos diferentes de vista, que definen diferentes modelos o líneas de negocio:

- Modelo P2P (*peer-to-peer*)
  La Figura 6.2 muestra el esquema seguido en el modelo de representación P2P. Este esquema está compuesto por tres nodos principales:

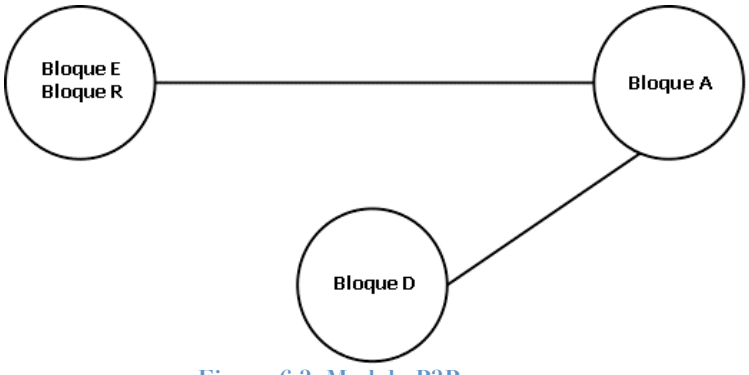

Figura 6.2: Modelo P2P

  - Bloque E/R:
    Este nodo es el encargado de monitorizar las señales del sistema para su posterior envío al bloque A. Una vez analizada la señal, se realizará una acción concreta sobre el sistema en función de la respuesta generada por el bloque A. Por último,

uno de los posibles roles que puede desempeñar este nodo es el de terminal móvil.

- Bloque A:
  Este nodo es el encargado de analizar la señal generada por el bloque E y generar una respuesta que será enviada al bloque R. Por último, uno de los posibles roles que puede desempeñar este nodo es el de ISP (*Internet Service Provider*) o empresa propietaria del terminal móvil.
- Bloque D:
  Este bloque es el encargado de proporcionar información al bloque A sobre las posibles amenazas que modifican el comportamiento normal del sistema. Por último, uno de los posibles roles que puede desempeñar este nodo es el de almacén de datos.

Por último, este esquema de representación permite una distribución equitativa de la carga computacional entre los distintos nodos y una alta disponibilidad del servicio global frente a posibles fallos del sistema.

- Modelo cliente-servidor
  La Figura 6.3 muestra el esquema seguido en el modelo de representación cliente-servidor. Este esquema está compuesto por dos elementos principales:

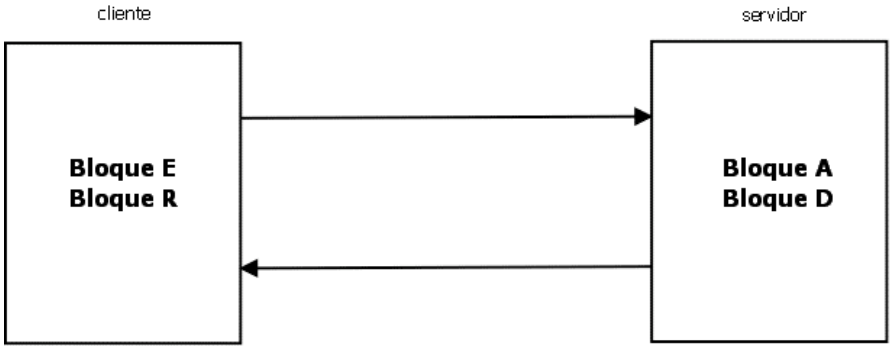

Figura 6.3: Modelo cliente-servidor

- Cliente:
  Este elemento es el encargado de monitorizar las señales del sistema y enviarlas al servidor correspondiente para su posterior análisis. Una vez analizada la señal, se realizará una acción concreta sobre el sistema en función de la respuesta generada por el servidor. Por último, el principal rol de cliente del sistema viene determinado por el terminal móvil.
- Servidor:
  Este elemento es el encargado de analizar las señalas generadas por el cliente mediante un proceso de comparación con las posibles amenazas del sistema almacenadas en contenedores de información. Por último, el principal rol de servidor del

sistema viene determinado por el ISP o empresa propietaria del terminal móvil.

Finalmente, este modelo permite un menor balanceo de carga respecto al caso anterior, pero con mayor sencillez y facilidad de implementación.

Tras la definición de la arquitectura principal del sistema, se procederá a la descripción detallada de cada uno de sus componentes, y los correspondientes detalles de implementación en la plataforma de desarrollo elegida durante la fase de análisis, *Windows Mobile*. El componente de monitorización de eventos del sistema o Bloque E es el elemento principal del dispositivo móvil, ya que siempre está alojado en el propio dispositivo con independencia del modelo de representación utilizado. La monitorización de eventos en dispositivos móviles consta de dos puntos de vista diferentes, un modelo intrusivo y otro no intrusivo.

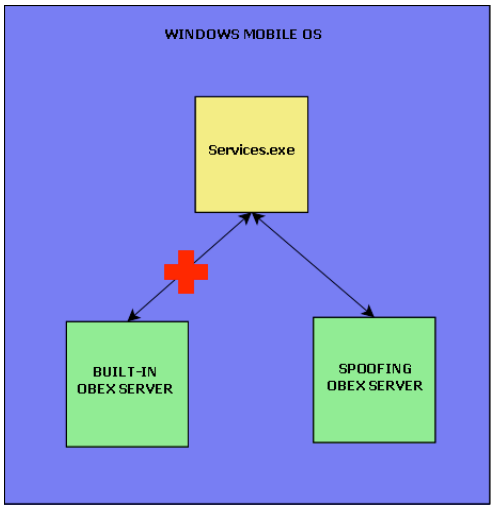

Figura 6.4: Modelo de monitorización intrusivo

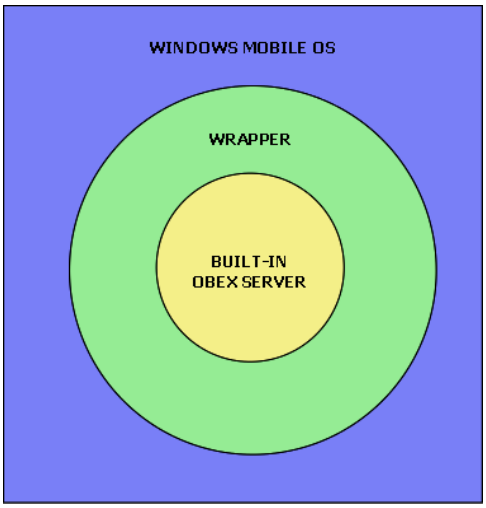

Figura 6.5: Modelo de monitorización no intrusivo

La Figura 6.4 muestra un modelo de representación intrusivo donde se reemplaza el servicio del sistema, o servicio *built-in*, por un servidor propio que monitorice las correspondientes peticiones. Este modelo conlleva ciertos riesgos asociados a la confiabilidad del servidor de reemplazo y sus correspondientes dificultades de implementación, pero permite a su vez un mayor control sobre las peticiones que llegan al dispositivo móvil. Por el contrario, la Figura 6.5 muestra un esquema de representación no intrusivo basado en la creación de un envoltorio o *wrapper*, que monitorice mediante manejadores o *handlers*, las diferentes peticiones que se realicen sobre este servicio. Este esquema posee ciertas ventajas como la ausencia de implementación de un nuevo servidor. Sin embargo, la plataforma de desarrollo seleccionada posee numerosas limitaciones en la implementación de este tipo de modelos de representación, por lo que no ha sido posible su implementación de manera efectiva. No obstante, la elección de otras plataformas puede evitar estas restricciones. Por último, las acciones

realizadas sobre los servicios, tanto desde el modelo intrusivo como desde el no intrusivo, requieren la ejecución de comandos nativos de entrada/salida IOCTL (*Input/Output Control*), implicando la utilización de DLL (*Dynamic Link Library*) nativas y su correspondiente desarrollo a través de la interfaz *P/Invoke* (*Platform Invoke*) proporcionada por .NET. Además, la ejecución de estos comandos permite el control y la monitorización de un gran número de servicios del sistema, como por ejemplo el servidor de OBEX (*Object Exchange*), utilizado por el protocolo de comunicaciones Bluetooth para el intercambio de ficheros y la visualización del sistema de ficheros.

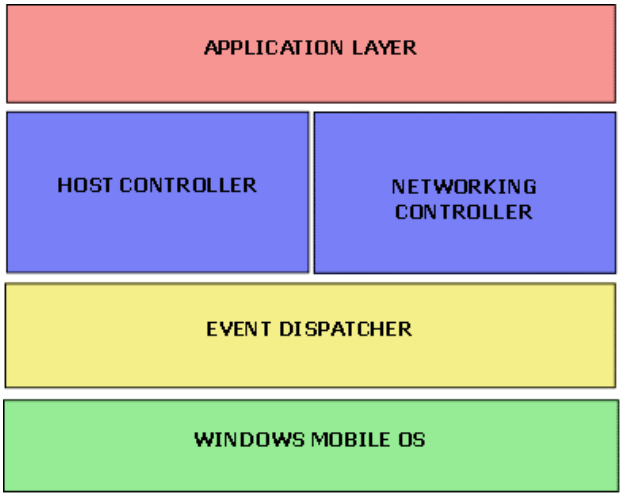

Figura 6.6: Arquitectura de monitorización propuesta

Tras la descripción de los esquemas de representación, la Figura 6.6 muestra la arquitectura propuesta, compuesta por dos niveles de monitorización de igual prioridad. En primer lugar, el nivel de monitorización *host* es el encargado de controlar y gestionar cualquier cambio realizado en las zonas protegidas del registro del sistema operativo, estableciendo un nivel de seguridad *registry data caging* a semejanza de otros sistemas móviles como *Symbian OS*. En segundo lugar, el nivel de *networking* es el encargado de controlar y gestionar los eventos generados por el protocolo Bluetooth y que están ampliamente descritos en esta sección.

El proceso de monitorización está basado en una arquitectura *WinSock Application* con *multithreading*, donde cada hilo abre un *socket* Bluetooth sobre un servicio concreto para su correspondiente escucha, y no finaliza hasta que, o bien se cierra la aplicación por completo, o bien el usuario decide finalizar este proceso mediante el apagado del Bluetooth del dispositivo. Por consiguiente, este sistema aporta una serie de ventajas como el análisis individualizado de cada uno de los servicios y una alta tolerancia frente a fallos, pero necesita de una mayor capacidad computacional y sobrecarga del sistema al existir un amplio rango de servicios a monitorizar. Además, las técnicas de monitorización empleadas bajo la plataforma de desarrollo *Windows Mobile* poseen una dificultad añadida, ya que los servicios del sistema o *built-in* no pueden ser monitorizados por lo que sólo

podrán emplearse técnicas no intrusivas si el servicio lo permite, o técnicas intrusivas por suplantación en el peor de los casos.
Posteriormente, tras la captura de un evento concreto, se formará un mensaje XML basado en los estándares IDMEF [51] e IDXP [52], que será enviado al componente de análisis o Bloque A, alojado generalmente en el servidor. La ubicación del servidor depende, directamente, de la arquitectura seleccionada, ya que en el caso de una arquitectura centralizada el servidor se encuentra dentro del propio terminal móvil, mientras que en el caso de una arquitectura distribuida el servidor se encuentra fuera del mismo. Este esquema, representado en la Figura 6.8, consta de una serie de atributos que tienen por objeto facilitar y estandarizar el intercambio y comunicación entre los diferentes módulos que componen el sistema de detección de intrusiones.

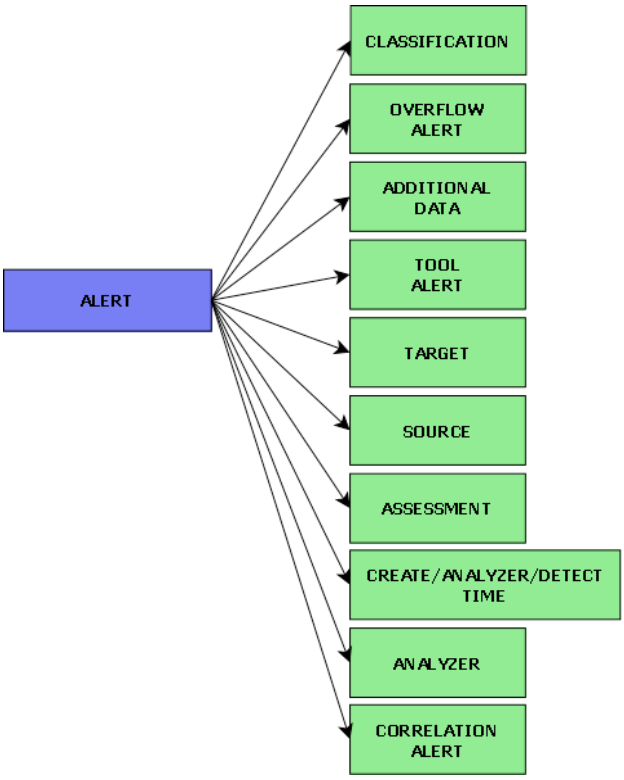

Figura 6.7: Esquema IDMEF [51]

Sin embargo, el esquema seguido en el sistema propuesto, utiliza únicamente una parte del conjunto total de IDMEF [51], al tratarse de un entorno de comunicación mas reducido respecto a los entornos de red habituales. Este esquema, representado en la Figura 6.9, diferencia dos tipos de mensajes distintos, petición o *request* y respuesta o *response*, determinados por atributos comunes y no comunes.

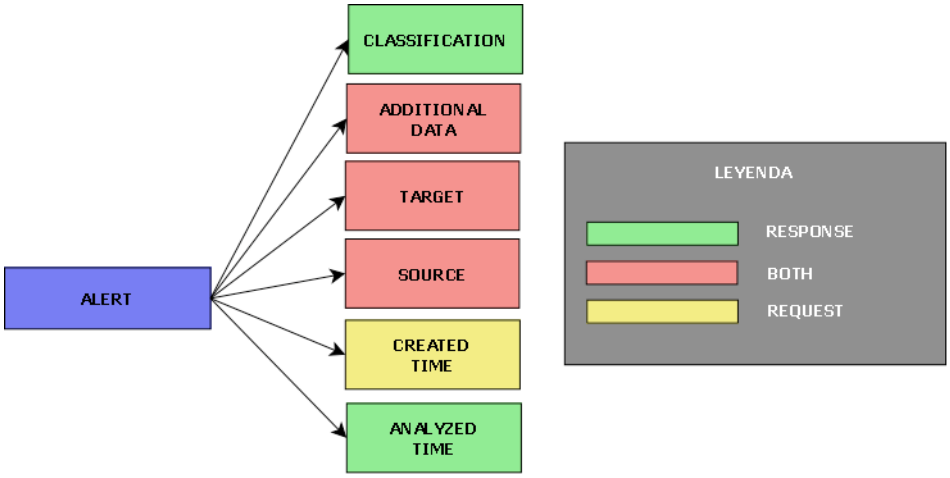

Figura 6.8: Esquema IDMEF de referencia

Tras la definición del esquema IDMEF [51] utilizado como referencia, se procederá a la descripción de cada uno de los campos, junto con sus respectivos atributos.

- Classification

  Este campo tiene por objeto describir el resultado del análisis realizado por el Bloque A. Los atributos que componen este campo son:

  - Id

    Este atributo indica el identificador único del evento analizado.
  - Title

    Este atributo indica el titulo asociado al evento analizado.
  - Impact

    Este atributo indica el grado de impacto del evento analizado.
- Additional Data

  Este campo tiene por objeto definir toda la información asociada al evento correspondiente, para que posteriormente sea analizada por el Bloque A. Generalmente, esta información se presentará mediante una estructura de datos XML, incluida dentro de un campo CDATA, y personalizada en función del perfil Bluetooth asociado al evento correspondiente.
- Source

  Este campo tiene por objeto definir el dispositivo que intenta establecer la comunicación con el dispositivo protegido. Los atributos que componen este campo son:

  - Name

    Este atributo indica el nombre del dispositivo, o la dirección física BD_ADDR en su defecto.
  - Decoy

    Este atributo determina la falsedad o veracidad de la dirección del dispositivo.
  - Service

    Este atributo identifica el perfil Bluetooth asociado al evento analizado.
- Target

Este campo tiene por objeto definir el dispositivo protegido. Los atributos que componen este campo son:

- Name

  Este atributo indica el nombre del dispositivo, o la dirección física BD_ADDR en su defecto.

- Spoofed

  Este atributo indica si el dispositivo puede estar suplantado o no.

- Service

  Este atributo identifica el perfil Bluetooth asociado al evento analizado.

- Created Time

Este campo tiene por objeto definir la marca de tiempo relacionada con la creación y posterior detección del evento analizado.

- Analyzed Time

Este campo tiene por objeto definir la marca de tiempo relacionada con el análisis del evento correspondiente.

El proceso de análisis, llevado a cabo por el Bloque A, está basado en los procedimientos de detección por firmas descritos en el estudio del estado del arte de los sistemas de detección actuales. Estos procedimientos están compuestos por dos elementos principales, un fichero de reglas y una base de datos relacional con las firmas de posibles ataques. Además, ambos elementos se encuentran en el componente de almacenamiento de la información o Bloque D, alojado generalmente el servidor o en un tercer componente confiable. El formato de los ficheros de reglas y la base de datos es el utilizado por la mayoría de los sistemas del mercado, con el objetivo de garantizar un nivel de interoperabilidad adecuado. Las Figuras 6.10 y 6.11 se corresponden con los esquemas de producción de reglas y firmas de ataques respectivamente.

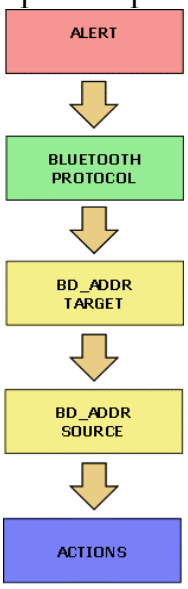

Figura 6.9: Modelo de representación de reglas propuesto

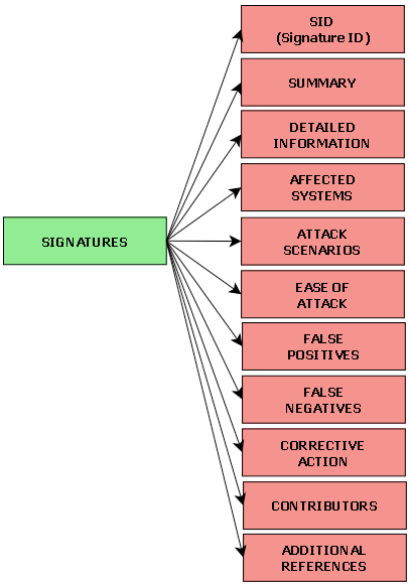

Figura 6.10: Modelo de representación de firmas propuesto

Finalmente, el Bloque A finalizará el proceso de análisis mediante el envío de un mensaje de respuesta IDMEF [51] al componente de respuesta o Bloque R, alojado generalmente en el dispositivo móvil, y prestando especial atención al nodo *classification*, que contiene los resultados del análisis del evento correspondiente. Una vez procesado este mensaje se aceptará o denegará la comunicación existente. No obstante, la realización de técnicas, tanto reactivas como proactivas más elaboradas, se estudiará en trabajos futuros.

6.2. MODELO DE DETECCIÓN ADAPTATIVO SOBRE BLUETOOTH

Tras la fase de diseño correspondiente al sistema de detección de intrusiones para terminales móviles, se va a proceder a la descripción de un modelo de detección adaptativo, que garantice una optimización de la carga computacional y un mejor rendimiento del sistema. Actualmente, la mayoría de los sistemas de detección de intrusiones conllevan altas cargas de trabajo y la monitorización de eventos a bajo nivel, que implica un proceso de instalación en ordenadores independientes y unos procedimientos complejos de comprobación e identificación de intrusiones. Por este motivo, la principal motivación para el diseño de este modelo consiste en la monitorización de niveles superiores que reduzcan la complejidad de las técnicas de identificación y comprobación, y su descomposición en función de las exigencias del entorno de ejecución. Sin embargo, este proceso de descomposición no es posible en todos los casos, ya que requiere protocolos de comunicación que lo permitan, como por ejemplo el protocolo de comunicación seleccionado para este modelo, Bluetooth.

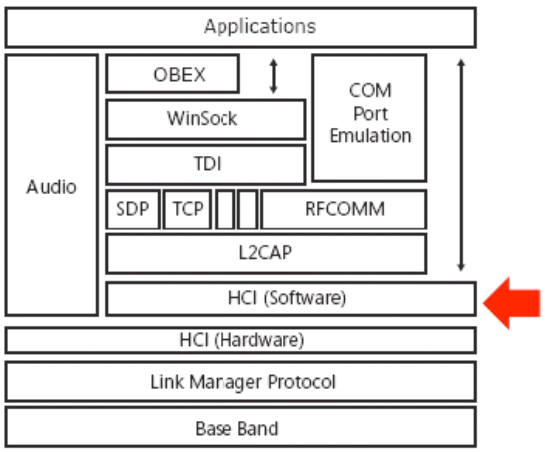

Figura 6.11: Arquitectura Bluetooth I

Tomando como referencia la pila de comunicaciones Bluetooth de *Microsoft*, la Figura 6.12 muestra la relación entre el nivel de monitorización de bajo nivel seguido en los actuales sistemas de detección y su correspondiente localización en la arquitectura Bluetooth. La monitorización de bajo nivel, indicado a través de la flecha adjunta al gráfico, requiere complejos

procedimientos de identificación de intrusiones, y soporta un elevado flujo de datos, que a fin de cuentas, empeora el rendimiento global del sistema. Además, otra de las principales desventajas de este tipo de monitorización en terminales móviles viene determinada por las propias restricciones del sistema operativo, ya que la gran mayoría deniegan el acceso a niveles de comunicación tan bajos sin ciertos privilegios, es decir, desde el punto de vista de la plataforma de desarrollo *Windows Mobile*, se encuentran en el modo de ejecución *kernel mode*.

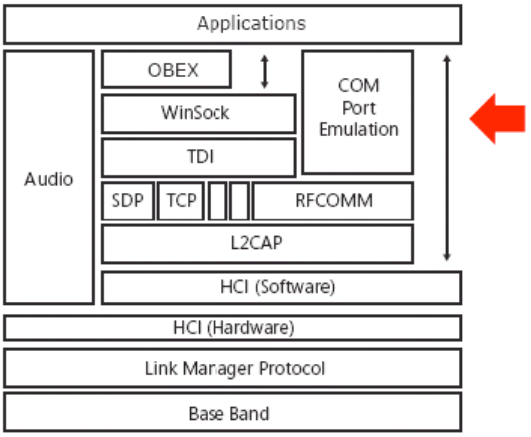

Figura 6.12: Arquitectura Bluetooth II

Por el contrario, la Figura 6.13 muestra la relación entre el modelo propuesto mediante la monitorización de alto nivel y su localización en la arquitectura Bluetooth. Este modelo permite descomponer el flujo de datos en función de los diferentes perfiles soportados por la arquitectura Bluetooth, y así, reducir la carga computacional de las técnicas de identificación y comprobación de intrusiones. Además, otras de las ventajas proporcionadas por este modelo son una personalización de las técnicas de identificación de intrusiones en función del perfil utilizado, que permita la reducción de falsos positivos y negativos, y una menor limitación de implementación.

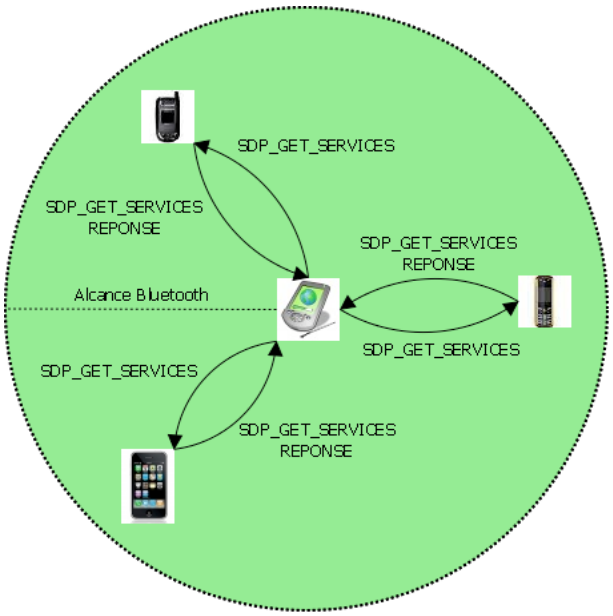

Figura 6.13: Modelo de Detección Adaptativo

La Figura 33 muestra el modo de funcionamiento adaptativo propuesto, donde el dispositivo móvil basado en el sistema operativo *Windows Mobile* y con el sistema de detección de intrusiones activo, realizará búsquedas periódicas de servicios en dispositivos cercanos mediante el protocolo de comunicaciones SDP (*Service Discovery Protocol*). Los principales aspectos a tener en cuenta durante su diseño son:

- Alcance de las búsquedas
  El alcance de las búsquedas viene determinado por el tipo de transceptor Bluetooth que contenga el dispositivo móvil. Generalmente, los módulos instalados son de clase 2 con un alcance aproximado de 15 metros.
- Periodicidad de las búsquedas
  La periodicidad de las búsquedas es parametrizable, debiendo siempre respetar la especificación Bluetooth y las limitaciones internas del dispositivo. Sin embargo, la elección de la periodicidad es un factor crítico durante la fase de diseño, ya que valores bajos pueden dan lugar a una sobrecarga del sistema, mientras que valores altos dan lugar a una pérdida de seguridad temporal crítica.
- Número máximo de dispositivos
  El número máximo de dispositivos es también parametrizable, aunque su valor depende directamente del tiempo de espera entre búsquedas y el tiempo de realización de la propia búsqueda, ambos parametrizables. Al igual que en el caso anterior, la asignación adecuada de estos valores es un factor crítico de diseño.

Tras la ejecución de cada búsqueda, se formará una lista con todos los servicios detectados, y se analizarán individualmente siguiendo el proceso de ejecución descrito en la sección anterior. Debido a esto, se podrán implementar analizadores orientados a un único flujo de datos más robustos y eficientes. El proceso de formación de listas de servicios es completamente

dinámico, ya que cada búsqueda habilita o deshabilita los analizadores correspondientes en función de los servicios detectados, permitiendo así, optimizar el rendimiento y la capacidad de almacenamiento del dispositivo móvil. Además, otra de las principales funcionalidades añadidas es la identificación del fabricante del dispositivo móvil mediante el valor OUI (*Organizacional Unique Identifier*) [40]. Este concepto, definido por el IEEE, se obtiene a partir de los tres primeros *bytes* de la dirección física del módulo Bluetooth (BD_ADDR), y permite, mediante la combinación con el valor *classOfDevice*, identificar posibles suplantaciones *BD_ADDR Spoofing* [41] o ataques *man-in-the-middle*. La Figura 6.15 muestra el esquema de detección de un ataque de suplantación común.

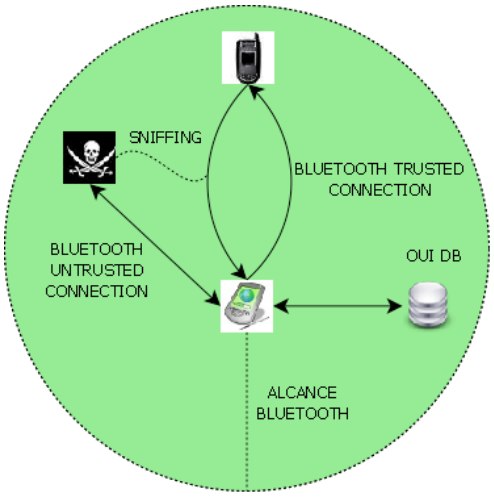

Figura 6.14: Esquema de detección por suplantación

Finalmente, el siguiente modelo está basado en la arquitectura descrita del apartado anterior bajo la plataforma de desarrollo Windows Mobile y la API de acceso Bluetooth 32feet [50].

6.3. SISTEMA PEP PARA TERMINALES MÓVILES

La última variante del sistema de detección propuesto consiste en el diseño de un sistema de políticas, orientado especialmente a entornos empresariales. Este sistema tiene por objeto gestionar y controlar las comunicaciones Bluetooth de un conjunto definido de dispositivos móviles mediante la ejecución de políticas alojadas o bien en un servidor externo, o en el propio dispositivo mediante bases de datos embebidas. Además, las técnicas de monitorización y análisis son análogas respecto a los procedimientos descritos en la primera sección, e incluso la estructura e información contenida en la base de datos puede ser reutilizada para este tipo de sistemas. Sin embargo, la principal diferencia radica en las técnicas de respuesta, ya que únicamente se aceptará o denegará la comunicación existente, en lugar de devolver respuestas más elaboradas. Por último, el modelo descrito está basado en la misma arquitectura y plataforma de desarrollo que en apartados anteriores.

6.4. CONCLUSIONES

El presente capítulo ha descrito los detalles de diseño del sistema propuesto, los diferentes modos de operación, variaciones de la arquitectura principal del sistema, y sus correspondientes limitaciones de implementación de la plataforma de desarrollo y el protocolo de comunicaciones seleccionados. Además, se han especificado las estructuras de datos utilizadas en la comunicación entre los diferentes módulos del sistema.
Por último, tras la especificación del proceso de análisis y realización de las fases de diseño e implementación se ha finalizado con la validación del sistema mediante la metodología VERDICT [34], utilizada para la realización auditorias de seguridad sobre sistemas de comunicación inalámbricos WLAN [1].

7. VALIDACIÓN DEL PROYECTO

Tras la fase de diseño, se ha procedido a la validación final del sistema, tanto teórica como práctica, que dé por finalizado la especificación del sistema descrito. En primer lugar, se ha realizado una breve introducción a la metodología de validación VERDICT [44], y que ha sido utilizado a posteriori por los estudios citados a continuación para la auditoria de seguridad del protocolo de comunicaciones Bluetooth. Además, se analizarán los resultados obtenidos de la evaluación de seguridad Bluetooth realizada por *Hager y Midkiff* [34] con las ventajas que aporta el sistema propuesto, y así, enumerar las ventajas finales que este sistema aporta. Por último, se ha realizado una comparativa final de rendimiento entre el sistema propuesto y un sistema de seguridad móvil en producción, *Kaspersky Mobile Security* [45].

7.1. METODOLOGÍA DE VALIDACIÓN VERDICT

La metodología VERDICT [44] consiste en una taxonomía de validación genérica sobre protocolos de comunicación inalámbricos creada por *Lough*. Este esquema describe un ataque como una secuencia de sucesos inadecuados ejecutados en el sistema, y realiza la validación del sistema en función de los siguientes parámetros:

- Validación (*Validation*)
- Exposición (*Exposure*)
- Aleatorización (*Randomness*)
- Liberación (*Deallocation*)

En primer lugar, el concepto de validación (*validation*) es el aspecto más problemático en cualquier protocolo de comunicación inalámbrico. Por este motivo, debe ser el elemento más evaluado mediante comprobaciones periódicas de las implementaciones existentes. Las principales comprobaciones a llevar a cabo durante la evaluación del sistema son:

- **Comprobación de la dirección física del dispositivo (BD_ADDR).**
  El siguiente proceso de comprobación está basado en la validación de la dirección física en función de las directivas del estándar IEEE 802.3 que evite cualquier tipo de suplantación.
- **Comprobación de estados inválidos en la gestión de claves de enlace.**
  Este proceso de comprobación está basado en la consideración de los posibles estados inválidos en la gestión de las claves de enlace y su correspondiente recuperación en el caso que éstos sean alcanzados.
- **Comprobación de estados inválidos en la gestión de los modos de encriptación.**

Este proceso de comprobación, al igual que el caso anterior, está basado en la consideración de los posibles estados inválidos en la gestión de los modos de encriptación y su recuperación.

- **Comprobación de las claves de encriptación.**
  Este proceso está basado en la comprobación de la robustez de las claves de encriptación mediante la validación del tamaño de clave establecido durante la negociación entre maestro y esclavo.
- **Comprobación de las claves de enlace.**
  Este proceso, al igual que el caso anterior, está basado en la comprobación de la robustez de las claves de enlace.

El proceso de validación tiene por objeto evitar desbordamientos del tipo *heap* y *buffer overflow*.

En segundo lugar, el concepto de exposición (*exposure*) determina la conectividad establecida por el dispositivo objetivo. Además, las principales comprobaciones realizadas son:

- **Prevención del intercambio de roles maestro esclavo.**
  Este proceso tiene por objeto prevenir el proceso de intercambio de roles entre maestro esclavo que reduzca la privacidad en las comunicaciones dentro de una red piconet.
- **Prevención del rastreo de las claves de enlace.**
  Este proceso está basado en prevenir la obtención de claves de enlace mediante técnicas de rastreo o *sniffing*.

El proceso de exposición tiene por obejto evitar las técnicas de rastreo o *sniffing* utilizadas en el *wardriving*, mediante la reducción de la exposición del dispositivo objetivo a los casos estrictamente necesarios.

El tercer parámetro utilizado durante la fase de validación es la aleatorización (*randomness*). Este parámetro determina la robustez de los algoritmos de generación de números pseudo-aleatorios y sus correspondientes semillas o *seeds*. Por este motivo, las principales comprobaciones realizadas son:

- **Comprobación de los algoritmos de generación de números pseudo-aleatorios.**
  Este proceso tiene por objeto validar la robustez de los algoritmos de generación de números pseudo-aleatorios.

El proceso de aleatorización tiene por objeto evitar la generación de códigos de seguridad débiles que faciliten los ataques por fuerza bruta.

Finalmente, el último de los parámetros utilizados durante la fase de validación es la liberación de recursos utilizados (*deallocation*). Este aspecto está basado en la eliminación de los recursos utilizados durante una comunicación inalámbrica, y está compuesto por las siguientes comprobaciones:

- **Liberación previa al intercambio de roles maestro esclavo.**
  Este proceso se complementa con la prevención del intercambio de roles entre maestro esclavo, ya que en el caso que éste se produzca,

deberá liberar todos los recursos establecidos en la red piconet, de tal manera que garantice la privacidad de todos los nodos de la red.

- **Liberación de las claves de enlace.**
  Este proceso tiene por objeto la eliminación de las claves de enlace establecidas durante una comunicación inalámbrica, para que no puedan ser utilizadas por cualquier intruso.
- **Comprobación de la ejecución del proceso de encriptación previo al proceso de autenticación.**
  Este proceso tiene por objeto la ejecución previa del proceso de encriptación para incrementar la seguridad del proceso de autenticación.

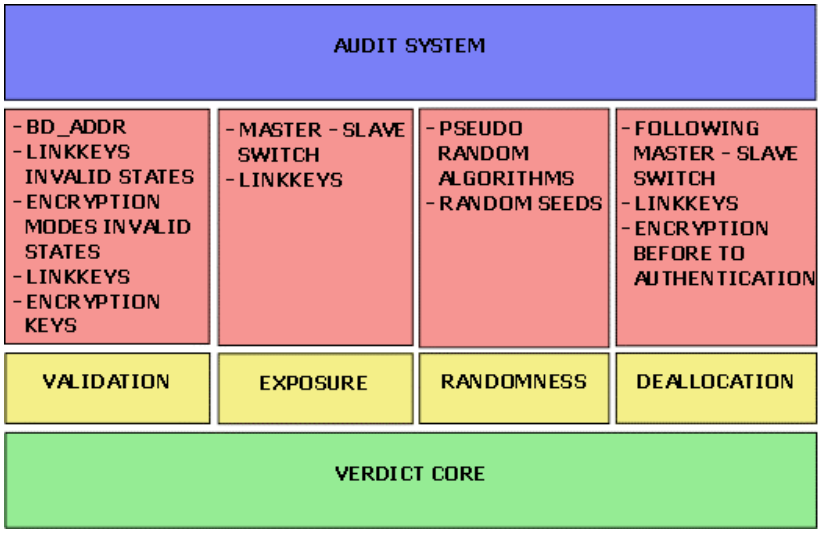

Figura 7.1: Arquitectura VERDICT [44]

Por último, el proceso de liberación tiene por objeto eliminar la mayoría de los recursos utilizados durante una comunicación inalámbrica, y así evitar que cualquier intruso pueda utilizarlos para establecer comunicaciones alternativas. La Figura 7.1 muestra un resumen gráfico de la metodología descrita.

7.2. VALIDACIÓN Y COMPARACIÓN TEÓRICA DEL SISTEMA

Tras la descripción del esquema de validación VERDICT, se procederá a la comparación entre los resultados obtenidos del estudio realizado por *Hager y Midkiff* [34], y aquellos obtenidos de la auditoria realizada sobre la especificación del sistema diseñado. El estudio de seguridad realizado por *Hager y Midkiff* sobre el protocolo de comunicaciones inalámbricas Bluetooth define algunas de las principales vulnerabilidades que este protocolo posee:

- Suplantación de las claves en enlace mediante ataques MiTM (*man-in-the-middle*).
- Suplantación de la dirección física BD_ADDR.

– Ataques por fuerza bruta sobre el código PIN.

En lo referente al estudio de seguridad realizado sobre el sistema diseñado, cabe destacar la ausencia de las vulnerabilidades por suplantación. En primer lugar, los casos por suplantación de las claves de enlace entre diferentes comunicaciones Bluetooth se encuentran solucionados ya que el sistema propuesto, dentro del controlador *host*, es capaz de eliminar las claves de enlace generadas durante una comunicación Bluetooth una vez que ésta ha finalizado. Sin embargo, los casos de suplantación durante la comunicación Bluetooth únicamente podrán ser solucionados si la acción realizada es capturada mediante las firmas instaladas en el sistema. En segundo lugar, los casos por suplantación de la dirección física BD_ADDR se ven reducidos en gran medida por las medidas preventivas que el sistema adaptativo proporciona basándose en el concepto del OUI (*Organizationally Unique Identifier*) [49] y el parámetro *classOfDevice*. Sin embargo, los ataques por fuerza bruta sobre los códigos PIN no son validados por el sistema especificado, siendo dirigido a los trabajos futuros del estudio dentro del apartado del controlador *host*. La Figura 7.3 muestra un resumen del proceso de validación realizado.

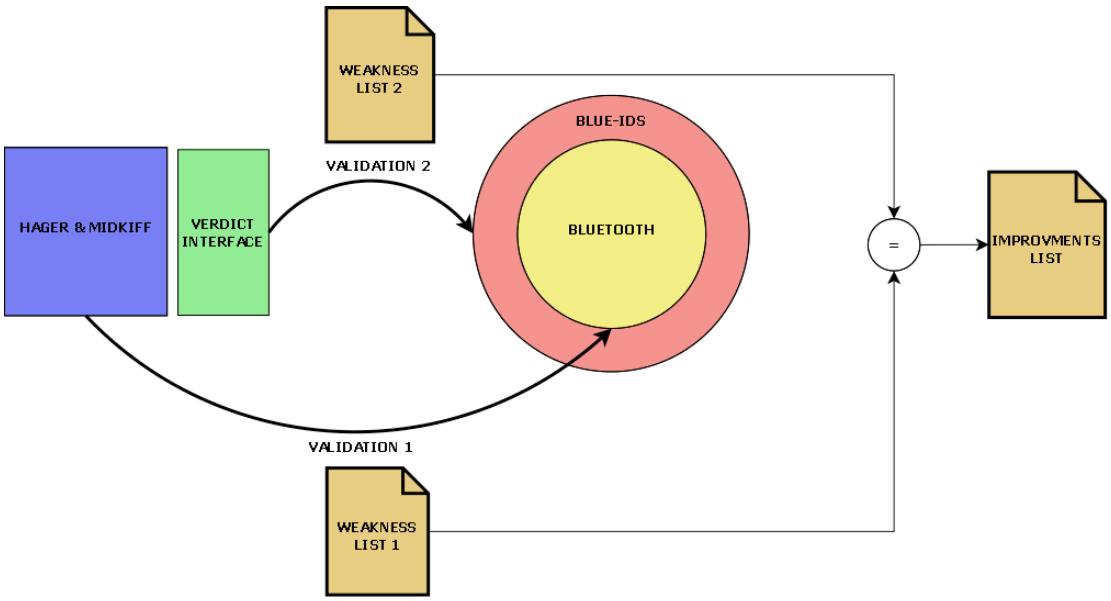

Figura 7.2: Esquema de Validación

7.3. VALIDACIÓN Y COMPARACIÓN PRÁCTICA DEL SISTEMA

Tras la validación teórica, se ha procedido a la evaluación práctica que permita evaluar el rendimiento del sistema, dentro de un terminal móvil *HTC P3300* con sistema operativo *Windows Mobile 6.0*. Además, se finalizará el estudio con una comparación final respecto a un sistema de seguridad en producción *Kaspersky Mobile Security* [45].

Este proceso de validación está compuesto por tres categorías diferentes, el consumo de almacenamiento, el nivel de procesamiento de la información y el uso de CPU. Por último, se realizará un breve estudio del sistema en producción mediante la evaluación de la capacidad de detección.
El consumo de almacenamiento viene determinada por la cantidad de memoria ROM, interna o mediante medios extraíbles (tarjetas SD), que ocupa la aplicación evaluada. La Figura 7.4 muestra un diagrama con la comparativa asociada a la capacidad de almacenamiento de los dos sistemas evaluados.

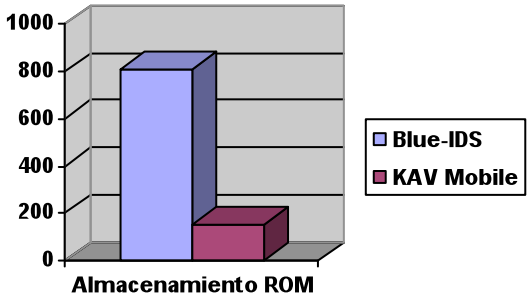

Figura 7.3: Validación del Consumo de Almacenamiento

Tal como se puede apreciar en la figura anterior, la capacidad almacenamiento del sistema en producción es significativamente más reducida que el sistema propuesto. Esto es debido, principalmente, a la distribución del almacenamiento total del sistema en producción, *KAV Mobile*, entre la memoria RAM y ROM, y al proceso de compilación con optimización de los recursos. Las líneas futuras de investigación relativas a la optimización del sistema propuesto tienen por objeto la distribución de la carga de almacenamiento en futuras versiones del sistema y a la compilación con optimización una vez que éste se encuentre concluído.
El nivel de procesamiento indica el número de hilos de ejecución que el sistema necesita para realizar las diferentes tareas requeridas. La Figura 7.5 muestra la comparativa asociada con el nivel de procesamiento entre ambos sistemas.

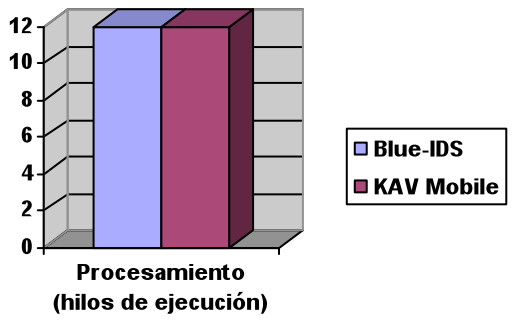

Figura 7.4: Validación del Nivel de Procesamiento

La figura descrita indica que el grado de procesamiento en hilos de ejecución entre ambos sistemas es equivalente.
El uso de CPU viene determinado por el porcentaje de tiempo que el procesador requiere para realizar las tareas implícitas del sistema o aplicación. La Figura 7.6 muestra la comparativa del uso de CPU entre ambos sistemas evaluados.

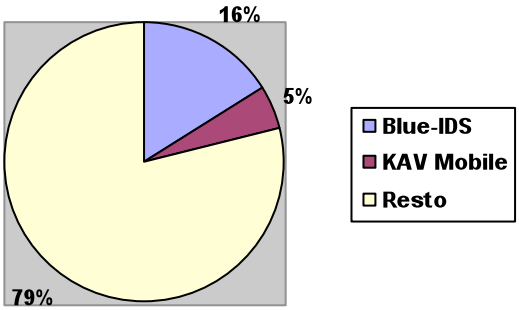

Figura 7.5: Validación del Uso de CPU

Tal como se puede apreciar en la figura descrita, el sistema propuesto realiza un mayor uso de CPU respecto al sistema en producción. Esta pequeña diferencia es debida, principalmente, al flujo de información que el sistema propuesto realiza entre los diferentes módulos que componen la arquitectura centralizada.
Finalmente, se ha realizado una pequeña evaluación del sistema en producción que dé por concluida la validación práctica del sistema propuesto. Este proceso de evaluación se realizará, fundamentalmente, en función de la capacidad de detección. La primera prueba de validación consistirá en la instalación de un antiguo virus diseñado para *Windows Compact Edition*, llamado *Duts* [46], a través de diferentes protocolos de comunicación, como por ejemplo Bluetooth o ActiveSync. Los resultados obtenidos del proceso de validación han sido satisfactorios, ya que ambas amenazas han sido detectadas y trasladadas a la bóveda de virus por petición del usuario. La Figura 7.7 y Figura 7.8 se corresponden con las capturas de pantallas del proceso de detección a través de ActiveSync y Bluetooth respectivamente.

Figura 7.6: Detección *Duts* con ActiveSync

Figura 7.7: Detección *Duts* con Bluetooth

La segunda y última prueba consistirá en la validación teórica de dos tipos de virus actuales diseñados para *Windows Mobile Compact Edition*, debido a la imposibilidad del proceso de instalación. Por este motivo, las descripciones detalladas de los virus analizados, *InfoJack* [47] y *FlexiSpy* [48] respectivamente, demuestran que la mayoría de los virus actuales en la plataforma analizada, son incluidos en las bases de datos de los sistemas en producción cada breves períodos de tiempo.
Por último, el proceso de validación realizado sobre el sistema en producción determina que dicho sistema ofrece unos niveles de seguridad muy satisfactorios, y una elevada usabilidad y accesibilidad sobre las diferentes acciones de análisis y configuración. Sin embargo, este sistema presenta ciertas debilidades ante ataques por suplantación (*BD_ADDR Spoofing*) y modificación del registro (*registry data caging*).

7.4. CONCLUSIONES

El presente capítulo ha consistido en dos bloques de validación independientes. En primer lugar, la validación teórica ha consistido en la comparación de los resultados obtenidos a partir de diferentes estudios de investigación, con aquellos obtenidos a partir de la auditoría VERDICT sobre el sistema propuesto. De igual manera, la validación práctica ha consistido en comparativas de rendimiento respecto a sistemas comerciales. Los resultados obtenidos del proceso de validación son indicadores de la precariedad de la seguridad móvil. Actualmente, existen una minoría de sistemas de antivirus focalizados en campos concretos. Por este motivo, la incorporación de sistemas como el propuesto, puede incorporar diferentes alternativas, que cubran amenazas y riesgos no resueltos por los sistemas actuales.

8. CONCLUSIONES

La creciente evolución que están experimentando los terminales móviles de última generación, está modificando la propia definición de teléfono móvil como un simple comunicador, hacia conceptos más avanzados como ordenadores personales de tamaño reducido. Por este motivo, y al igual que la mayoría de los ordenadores personales, las salvaguardas proporcionadas por la propia arquitectura de seguridad del sistema operativo, son insuficientes en la prevención de las amenazas existentes. Además, los resultados obtenidos de la participación en congresos internacionales de seguridad como Secrypt [42] o RAID (*Recent Advances in Intrusion Detection*) [43], junto con el estudio realizado por TrendMicro [58], son indicadores de la precariedad de la seguridad móvil, siendo necesaria la creación de nuevos diseños e implementaciones de seguridad que permitan contrarrestar esta problemática.

Tal como se comentó en el apartado introductorio, la mayoría de las implementaciones existentes se corresponden con sistemas de antivirus optimizados para terminales móviles, por lo que la aparición de nuevos diseños de seguridad orientados a dispositivos móviles, entre los que se encuentra el sistema propuesto, tiene por objeto plantear diversas alternativas, y resolver las principales debilidades, de dichos sistemas. Sin embargo, problemáticas como la fragmentación móvil, dificultan la interoperabilidad del código fuente generado entre los diferentes sistemas operativos móviles existentes, debido a la fuerte dependencia entre el sistema propuesto y el sistema operativo móvil seleccionado. Por este motivo, una vez seleccionado el sistema operativo, una de las principales dificultades encontradas durante la implementación del sistema propuesto ha sido las restricciones proporcionadas por las diferentes API del sistema operativo, y en especial, en la administración y control de los servicios del sistema. No obstante, la elección del sistema operativo puede contribuir a la facilidad o dificultad de implementación del sistema propuesto.

Las variaciones asociadas al modelado del sistema propuesto indican que la implementación de herramientas de seguridad móvil posee una elevada proyección. Esto es debido a que la inclusión de protocolos de comunicación como Bluetooth [35] o *Wi-Fi* [1], generan una elevada cantidad de casos de uso sobre los cuales se permite implementar diferentes aplicaciones de seguridad. Por consiguiente, el sistema propuesto tiene por objeto la introducción de un nuevo modelo de sistema de detección de intrusiones ligero, como una herramienta de seguridad orientada a teléfonos móviles, que permita garantizar unos niveles de interoperabilidad y privacidad de la información de acuerdo con las actuales exigencias de los usuarios.

8.1. TRABAJOS FUTUROS

Por último, las líneas de investigación futuras van dirigidas en función de cuatro conceptos diferentes, diseño e implementación de herramientas de seguridad dirigidas por modelos, optimización en el rendimiento del sistema propuesto, introducción de modelos matemáticos de detección de anomalías, y optimización en la generación de firmas de ataques. En primer lugar, la incorporación de plataformas dirigidas por modelos, como por ejemplo MDA *(Modelo-Driven* Architecture) [59], permitirá contrarrestar los efectos negativos producidos por la fragmentación móvil. En segundo lugar, el proceso de optimización de rendimiento del sistema propuesto permitirá alcanzar el nivel alcanzado por los sistemas comerciales y garantizar una mejor experiencia de usuario. En tercer lugar, la introducción de modelos matemáticos permitirá comparar la capacidad de detección de los sistemas de detección de intrusiones basados en firmas, respecto a los sistemas basados en anomalías, con el objetivo de dejar aquel con mejores resultados. En cuarto, y último lugar, el proceso de optimización de firmas permitirá reducir el tiempo de procesamiento de *matching*.

9. REFERENCIAS

1. IEEE 802.11 Standard Specification. http://standards.ieee.org/getieee802/802.11.html
2. IEEE 802.16 Standard Specification. http://standards.ieee.org/getieee802/802.16.html
3. OMTP, Open Mobile Terminal Platform. http://www.omtp.org/
4. Symbian Software Ltd., http://www.symbian.com/index.asp
5. Microsoft Windows Mobile, http://www.microsoft.com/spain/windowsmobile/5/default.mspx
6. Apple Inc., http://developer.apple.com/iphone/
7. Research In Motion Ltd., http://www.blackberry.com/
8. Palm Inc., http://www.palm.com/
9. Google Android, http://code.google.com/intl/en/android/
10. Heath, C., 2006, Symbian OS Security Platform, Wiley.
11. LG Electronics. http://www.lge.com
12. Savoldi, A., Gubian, P., 2008, Symbian Forensics: An overview. IIHMSP '08.
13. Moreno, A., 2008, Windows Mobile Security Model, http://www.seguridadmobile.com/windows-mobile/seguridad-windows-mobile/ejecucion-aplicaciones.html
14. Microsoft, 2007, Security Model for Windows Mobile 5 and 6.
15. Apple Inc., 2008, Security Overview.
16. Apple Inc., 2007, iPhone Reference Library. http://developer.apple.com/iphone/library/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/IPhoneOSOverview/IPhoneOSOverview.html
17. Research In Motion Ltd., 2008, Blackberry Enterprise Solution: Technical Overview.
18. Mitch Allen, 2009, Palm WebOS: Rough Cuts Version. O'Reilly.
19. Kingpin, Mudge, 2001, Security analysis of the palm operating system and its weaknesses against malicious code threats. Proceedings of the 10th conference on USENIX Security Symposium.
20. Gostev, A., 2008, Introducción a la virología móvil. Hakin9.
21. F-Secure Corp., 2009, F-Secure Mobile Detection Descriptions. http://www.f-secure.com/v-descs/mobile-description-index.shtml
22. Weaver, N., Paxson, V., Staniford, S., Cunningham, R., 2003, A Taxonomy of Computer Worms. Proceedings of the 2003 ACM Workshop on Rapid Malcode WORM.
23. OMTP, Open Mobile Terminal Platform, 2008, Security threads on embedded consumer devices.
24. Open Handset Alliance, 2009, Android Security Overview. http://developer.android.com/guide/topics/security/security.html

25. Bace, R. Intrusion Detection. 2000, Macmillan Technical Publishing.
26. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E., 2009, Anomaly-based network intrusion detection: Techniques, systems and challenges. Computer & Security.
27. CIDF, Common Intrusion Detection Framework. http://gost.isi.edu/cidf/
28. Ditcheva, B., Fowler, L., 2005, Signature-based Intrusion Detection.
29. Orebaugh, A., Biles, S., Babbin, J., 2005, Snort Cookbook. O'Reilly.
30. Neumann, P., Valdes, A., Lunt, T., Jagannathan, R., Lee, R., Listgarten, S., S., Edwards, D., Javitz, H., 1988, IDES: The Enhanced Prototype.
31. Kim, G., Spafford, E., 1995, The Design and Implementation of Tripwire: A File System Integrity Checker.
32. Zaraska, K., 2003, Prelude IDS: current state and development perspectives.
33. PreludeIDS Technologies, 2009, http://www.prelude-ids.com/en/welcome/index.html
34. Hager, Creighton T., Midkiff, Scott F., 2003, An Analysis of Bluetooth Security Vulnerabilities.
35. IEEE 802.15.1 Standard Specification. http://standards.ieee.org/getieee802/802.15.html
36. Bluetooth SIG. https://www.bluetooth.org/
37. Moreno, A., 2008, Bluetooth Specification, http://www.seguridadmobile.com/bluetooth/especificacion-bluetooth/perfiles-bluetooth/index.html
38. Kammer, D., McNutt, G., Senese, B., Bray, J., 2002, Bluetooth. Application Developer's Guide: The Short Range Interconnect Solution. Syngress.
39. Hager, Creighton T., Midkiff, Scott F., 2003, An Analysis of Bluetooth Security Vulnerabilities. IEEE Wireless Communications and Networking.
40. OUI Specification. IEEE. http://standards.ieee.org/regauth/oui/index.shtml
41. Moreno, A., 2008, BD_ADDR Spoofing Description, http://www.seguridadmobile.com/bluetooth/seguridad-bluetooth/BD_ADDR-spoofing.html
42. Secrypt, International Conference on Security and Cryptography. http://secrypt.icete.org/
43. RAID, Recent Advances on Intrusion Detection, http://www.rennes.supelec.fr/RAID2009/
44. Lough, Daniel Lowry, 2001, A Taxonomy of Computer Attacks with Applications to Wireless Networks.
45. Kaspersky Lab, 2008, Kaspersky Mobile Security, http://www.kaspersky.com/kaspersky_mobile_security.
46. Viruslist.com, 2004, Duts.A Virus Specification, http://www.viruslist.com/en/viruslist.html?id=1874404

47. F-Secure, 2009, InfoJack Virus Specification, http://www.f-secure.com/v-descs/trojan_wince_infojack.shtml
48. F-Secure, 2009, FlexSpy Virus Specification. http://www.f-secure.com/sw-desc/riskware_wince_flexispy_a.shtml
49. IEEE, OUI Databases, http://standards.ieee.org/regauth/oui/
50. In The Hand Ltd., 2009, 32feet Windows Mobile API. http://inthehand.com/
51. IETF, IDMEF: RFC 4765, 2007, http://www.ietf.org/rfc/rfc4765.txt
52. IETF, IDXP: RFC 4767, http://www.rfc-archive.org/getrfc.php?rfc=4767
53. Wikipedia, 2009, Mobile Operating Systems. http://en.wikipedia.org/wiki/Mobile_operating_system
54. Open Mobile Alliance, 2009, http://www.openmobilealliance.org/
55. Retamosa, G., López de Vergara, J., *Assessment of Mobile Security Platforms*, Secrypt, 2009.
56. Natural Science Foundation, Bro IDS, http://bro-ids.org/.
57. Trend Micro Inc., 2008, OSSEC. http://www.ossec.net/
58. Trend Micro Inc., 2009, Smartphone Report, *http://trendmicro.mediaroom.com/file.php/96/Trend+Smart+Smartphone+Report.ppt*
59. OMG (*Object Management Group*), 2009, *MDA, Model-Driven Architecture*. http://www.omg.org/mda/

10. GLOSARIO

ACRÓNIMOS